Dissertation

Advanced Algorithms and Software for Primary Vertex Reconstruction and Search for Flavor-Violating Supersymmetry with the ATLAS Experiment

Dissertation
zur Erlangung des Grades

Doktor der Naturwissenschaften

am Fachbereich Physik, Mathematik und Informatik
der Johannes Gutenberg-Universität
in Mainz

JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

Bastian Schlag
geboren in Hachenburg

Mainz, 2022

Datum der mündlichen Prüfung: 20. September 2022

1. Gutachter: - Aus datenschutzrechtlichen Gründen entfernt -
2. Gutachter: - Aus datenschutzrechtlichen Gründen entfernt -

Kurzfassung

Unser derzeit bestes Verständnis des Universums wird durch das Standardmodell der Teilchenphysik beschrieben, eine Theorie, die sich als bemerkenswert erfolgreich bei der Erklärung und Vorhersage vieler verschiedener physikalischer Phänomene, neuer Teilchen und ihrer Wechselwirkungen erwiesen hat. Trotz seines großen Erfolges ist das Standardmodell unvollständig, da es keine Erklärungen für verschiedene experimentelle Beobachtungen, wie z. B. die überwältigenden Beweise für dunkle Materie, liefern kann, was umfangreiche Forschungsprogramme an zukünftigen Teilchenbeschleunigern motiviert, um die Physik jenseits des Standardmodells besser verstehen und somit unser grundlegendes Verständnis des Universums neu gestalten zu können.
Die Rekonstruktion von Teilchenspuren und den dazugehörigen Produktionspunkten (*Vertices*) ist eine entscheidende Komponente in der Ereignisrekonstruktion fast aller Teilchenbeschleuniger-Experimente. Um die Leistung derzeitiger Spurrekonstruktions- und Vertexing-Algorithmen unter den bevorstehenden rechenintensiven Herausforderungen steigender Energien und immer größerer Luminositäten an zukünftigen Teilchenbeschleunigern aufrechterhalten oder sogar verbessern zu können, sind umfangreiche Algorithmus- und Software-Verbesserungen unerlässlich.
In dieser Arbeit werden eine völlig neue, experimentunabhängige, thread-sichere und hochleistungsfähige Vertex-Rekonstruktionssoftware sowie neuartige Algorithmen für die primäre Vertex-Rekonstruktion in Kollisionsumgebungen von Teilchenbeschleunigern mit hohen Luminositäten vorgestellt. Nach der Entwicklung und speziellen Leistungsoptimierung wurde die neu entwickelte Software vollständig in das Rekonstruktionssoftware-Framework des ATLAS Experiments am Large Hadron Collider (LHC) integriert und validiert. Aufgrund der herausragenden Physik- und CPU-Leistung wird die im Rahmen dieser Arbeit entwickelte Vertexing-Software als primäre Vertex-Rekonstruktionssoftware des ATLAS Experiments für den LHC Run 3 und darüber hinaus eingesetzt werden.
Des Weiteren wird unter Verwendung des vollständigen ATLAS Run 2 Datensatzes von $139\,\text{fb}^{-1}$ Proton-Proton-Kollisionsdaten, die bei einer Schwerpunktsenergie von 13 TeV am LHC aufgezeichnet wurden, eine Suche nach Flavor-verletzender Supersymmetrie, einer theoretischen Erweiterung des Standardmodells, die viele seiner derzeitigen Limitierungen bewältigen kann, vorgestellt. Die Suche richtet sich nach einem asymmetrischen 0-Leptonen Endzustand mit einem Top-Quark, einem Charm-Quark und großem fehlenden Transversalimpuls und stellt eine erste Analyse dieser Art in ATLAS dar. Es wurde kein signifikanter Überschuss zur Standardmodellvorhersage gefunden und beobachtete (erwartete) Ausschlussgrenzen von bis zu 880 GeV (1020 GeV) für Massen des supersymmetrischen Top-Quark Partners in einem Flavor-verletzenden supersymmetrischen Modell ermittelt, wodurch ein bisher weitgehend unerforschter Bereich des Parameterraums der Supersymmetrie ausgeschlossen werden konnte.

Abstract

Our best current understanding of the universe is given by the Standard Model of particle physics, a theory that has proven to be remarkably successful in explaining and predicting many different physics phenomena, new particles and their interactions. Despite its great success, the Standard Model is known to be incomplete as it cannot provide explanations for various experimental observations, such as the overwhelming evidence for dark matter, motivating extensive research programs at future particle colliders to better understand physics beyond the Standard Model and thus reshape our most fundamental understanding of the universe.

The reconstruction of charged particle trajectories (tracks) and their associated production vertices is a crucial component in the event reconstruction of almost all particle collider experiments. Extensive algorithmic and software improvements will be essential in order to maintain or even improve the performance of current track and vertex reconstruction algorithms under the upcoming computationally intensive challenges of ever increasing energies and luminosities at future particle colliders.

This thesis introduces an entirely new, experiment-independent, thread-safe and highly performant vertex reconstruction software suite as well as novel algorithms for primary vertex reconstruction in high-luminosity environments. After its development and dedicated performance optimization, the newly developed software has been fully integrated and validated in the reconstruction software framework of the ATLAS experiment at the Large Hadron Collider (LHC). Due to its outstanding physics and CPU performance, the vertexing software developed in the course of this thesis will be used as the default ATLAS primary vertex reconstruction tool for LHC Run 3 and beyond.

Furthermore, using the full ATLAS Run 2 dataset of $139\,\mathrm{fb}^{-1}$ proton-proton collision data recorded at a center-of-mass energy of 13 TeV at the LHC, a search for flavor-violating supersymmetry, a theoretical extension of the Standard Model that is able to overcome many of its current limitations, is presented. The search targets an asymmetric 0-lepton final state with a top quark, a charm quark and large missing transverse momentum and represents a first of its kind analysis in ATLAS. No significant excess over the Standard Model prediction was found and observed (expected) exclusion limits of up to 880 GeV (1020 GeV) for the supersymmetric top quark partner mass in a flavor-violating supersymmetric model were obtained, excluding a previously largely unexplored region of supersymmetry parameter space.

Contents

Für meine Eltern.

Chapter 1

Introduction

Our best current understanding of particle physics is given by the Standard Model, a theory that successfully explains a wide range of experimental results and precisely predicted many different physics phenomena and new particles, such as the Higgs boson discovered in 2012 by the ATLAS and CMS collaborations at CERN. Despite its remarkable success, the Standard Model is known to be incomplete as it cannot explain a number of experimental observations, such as the overwhelming astrophysical and cosmological evidence for dark matter, a non-baryonic form of matter we have yet to understand, which accounts for the vast majority of mass in our universe.

In the past decades, many theories for new physics beyond the Standard Model, such as Supersymmetry, emerged and provided well-motivated and promising theoretical frameworks to extend our fundamental understanding of particle physics and improve upon the shortcomings of the Standard Model. However, none of the numerous searches for new physics signatures that have been conducted at particle colliders in the past years could provide any direct evidence for the existence of new particles or forces as predicted by these theories.

Extensive research programs will be required in the future in order to better understand physics beyond the Standard Model and thus to reshape our most fundamental understanding of the universe. The Large Hadron Collider (LHC) at CERN will therefore undergo major upgrades towards very high instantaneous luminosities, aiming to further extend its great discovery potential in the future. The ever-increasing energies and luminosities will result in collision events of unprecedented complexity as well as immensely high data rates to be processed. Major software and computing upgrade efforts will be inevitable to be able to cope with these resource-intensive and computationally challenging conditions and to sustain or improve upon the current physics performance achieved at the LHC experiments, while staying within the budget limits and available computational resources. One of the most important aspects of these software and computing upgrade efforts is the development of novel algorithms and efficient software components for the reconstruction of collision events which need to be highly optimized for the usage in these complex environments.

An important building block in the reconstruction of collision events at particle colliders is the precise position determination of the primary interaction vertices of the colliding particles (the so-called primary vertex reconstruction), which is required to fully understand the complete kinematics of the physics processes and serves as an important input to almost all downstream reconstruction algorithms and physics analyses.
The first goal of this thesis is to develop an entirely new, state-of-the-art primary vertex reconstruction software suite, designed for parallel code execution and highly optimized in terms of its per-thread CPU performance, as well as new primary vertex reconstruction algorithms for high-luminosity environments. While the software's design should be flexible enough to be used by a variety of different experiments in high energy physics (HEP), the ultimate goal is to provide a full production-ready integration in the event reconstruction framework of the ATLAS experiment at the LHC, replacing the previous ATLAS primary vertex reconstruction software for the upcoming LHC Run 3 and beyond.
Additionally, a search for flavor-violating supersymmetry using the full ATLAS Run 2 dataset, comprising $139\,\mathrm{fb}^{-1}$ of proton-proton collision data recorded at a center-of-mass energy of 13 TeV at the LHC, is presented. Recent theoretical developments indicated that previous supersymmetry searches have not been sensitive to supersymmetric models in which the second and third generation of supersymmetric particles are allowed to undergo a significant mixing and proposed a dedicated search in a mixed-flavor final state with a top quark, a charm quark and large missing transverse energy. The presented search is the first of its kind and targets a fully-hadronic top decay mode, resulting in a 0-lepton final state that requires the simultaneous application of bottom and charm quark flavor tagging techniques.

This thesis is organized in three parts:
An introduction to theoretical concepts and experimental techniques is given in Part I.
Part II then describes state-of-the-art algorithms for the reconstruction of charged particle trajectories and primary vertices, the development of novel algorithmic approaches for primary vertex reconstruction in the upcoming high-luminosity environments as well as the development, implementation and application of an entirely new primary vertex reconstruction software suite.
Finally, Part III first gives an introduction to experimental and statistical methods used in searches for new physics and afterwards presents the ATLAS Run 2 search for flavor-violating supersymmetry in a 0-lepton final state with a top quark, a charm quark and large missing transverse energy.

Part I

Introduction to Theoretical Concepts and Experimental Techniques

Chapter 2

Theoretical Concepts in Particle Physics

2.1 The Standard Model of Particle Physics

The Standard Model (SM) of particle physics is a renormalizable quantum field theory that describes elementary particles and their fundamental interactions with the exception of gravity [1, 2]. It combines the theories of strong, electromagnetic and weak interaction as well as the Higgs mechanism and has been remarkably successful in both describing experimental results with very high precision as well as predicting the existence of fundamental particles, the last of which was observed in 2012 at the Large Hadron Collider at CERN.
A short overview of its most important concepts and properties will be given in this section. Note that, unless stated otherwise, the particle physics convention of natural units with $c = \hbar = 1$ will be used in the following.

2.1.1 Fundamental Interactions and Particle Content

The Standard Model is based on the principle of local gauge invariance with an underlying $SU_C(3) \times SU_L(2) \times U_Y(1)$ symmetry group. $SU_C(3)$ denotes the gauge group of quantum chromodynamics (QCD), the theory of strong interaction (see Section 2.1.2). The physics of strongly interacting particles that are charged under $SU_C(3)$ with the so-called *color charge* C is invariant under $SU_C(3)$ gauge transformations. The eight generators of the $SU_C(3)$ group correspond to the massless spin-1 force carriers of the strong interaction, the *gluons* g.
The electroweak sector of the SM (see Section 2.1.3) is represented by the $SU_L(2) \times U_Y(1)$ symmetry group, where the subscript L indicates that only left-handed particles are charged under $SU_L(2)$ and the subscript Y denotes the so-called *weak hypercharge*. The four generators of $SU_L(2) \times U_Y(1)$ correspond to the four massless gauge bosons W^1, W^2, W^3 and B, which allow to form the charged and neutral massive force mediators of the weak interaction, the $\mathrm{W}^{\pm}$ and Z^0 bosons, as well as the massless force carrier of the electromagnetic force, the photon γ, by spontaneous symmetry breaking and the Higgs mechanism (see Section 2.1.4).

As illustrated in Fig. 2.1, the SM additionally comprises twelve spin-1/2 matter particles, called *fermions*, which are grouped into three generations.

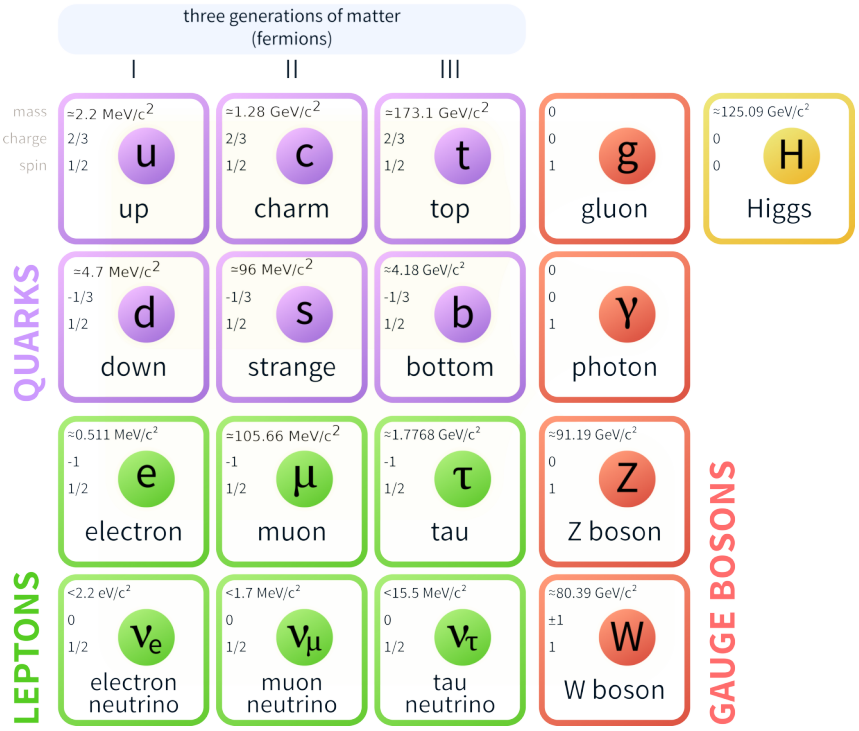


Figure 2.1: Summary of elementary particles in the Standard Model [3].

Fermions can be divided into two different types, *quarks* and *leptons*, depending on the kind of interactions they participate in. While quarks interact via all three fundamental forces described by the SM, leptons do not carry color charge and therefore only interact weakly as well as via the electromagnetic force if they are electrically charged.

Quarks come in six different *flavors*, called up (u), down (d), charm (c), strange (s), top (t) and bottom (b), and have fractional electric charge values of $+\frac{2}{3}e$ for up-, charm- and top-quarks (collectively referred to as up-type quarks) and $-\frac{1}{3}e$ for down-, strange- and bottom-quarks (down-type quarks). Due to the color confinement nature of QCD (see Section 2.1.2), color-charged particles such as quarks and gluons cannot exist as free particles but instead form colorless bound states, called *hadrons*, that can be observed experimentally.

Leptons can be divided into electrically charged and neutral flavors. The three electrically charged lepton flavors, the electron e, muon μ and tau τ, come with an electric charge of $-1e$, while their corresponding neutrino flavors, the electron neutrino ν_e, muon neutrino ν_μ and tau neutrino ν_τ, do not carry electric charge and thus only interact via the weak force.

For each of the above described twelve fermions, the SM contains a corresponding antiparticle with the exact same properties but opposite charges.

The list of known elementary particles in the SM is completed by the *Higgs boson*, a manifestation of the Higgs field which is responsible for generating the masses of all massive particles in the SM [4].

2.1.2 Quantum Chromodynamics

The QCD Lagrangian describes interactions of quark fields ψ_i, with color index i and mass m, and gluon fields A^a_μ $(a = 1, \ldots, 8)$ and is given by

$$\mathcal{L}_{\mathrm{QCD}} = -\frac{1}{4} G^{a\mu\nu} G^a_{\mu\nu} + \bar{\psi}_i (i\gamma^\mu D_\mu - m)\psi_i, \tag{2.1}$$

where $G^a_{\mu\nu}$ denotes the field strength tensor defined as

$$G^a_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\nu A^a_\mu + g_s f^{abc} A^b_\mu A^c_\nu \tag{2.2}$$

with coupling constant g_s and structure constant of the gauge group f^{abc}. In order to ensure gauge invariance of $\mathcal{L}_{\mathrm{QCD}}$, the covariant derivative D_μ is defined as

$$D_\mu = \partial_\mu - \frac{i}{2} g_s t^a A^a_\mu \tag{2.3}$$

with the 3×3 Gell-Mann matrices t^a, the generators of $SU_C(3)$ [4, 5].
The first term in Eq. (2.1) together with the field strength tensor definition given in Eq. (2.2) generate non-linear terms of the form $g_s f^{abc}(\partial^\mu A^{a\nu} - \partial^\nu A^{a\mu}) A^b_\mu A^c_\nu$ as well as $g_s^2 f^{abc} f^{ade} A^{b\mu} A^{c\nu} A^d_\mu A^e_\nu$, giving rise to three-gluon and four-gluon self-interaction vertices, respectively. These self-interactions of the QCD force mediators lead to interesting properties of the strong force that will be briefly discussed in the following.
Similar to quantum electrodynamics (QED), where vacuum polarization caused by virtual electron-positron pairs leads to partial screening of a bare electric charge and therefore to an effective charge that becomes smaller with larger distances, the existence of virtual quark-antiquark pairs in QCD can result in partial color charge screening effects. However, gluon self-interaction vertices in QCD allow also for virtual gluon-gluon pairs to be created from the vacuum, resulting in a gluon cloud that spatially dilutes the initial bare color charge. This effect of QCD *anti-screening* typically dominates over screening effects from quark-antiquark loops and leads to the strong coupling constant $\alpha_s(Q^2) = \frac{g_s^2(Q^2)}{4\pi}$ becoming smaller at smaller distances (larger energy scales Q) and therefore gives rise to the QCD property of *asymptotic freedom*: quarks and gluons become asymptotically free and behave as free particles at very short distances, allowing perturbative QCD approaches with sufficiently small couplings [6].
At larger distances, or lower energy scales Q, on the other hand, the coupling becomes increasingly stronger, which makes the isolation of a single quark or gluon impossible, a QCD phenomenon referred to as *color confinement*. Due to QCD color confinement, single quarks and gluons form color-neutral hadrons in a process called *hadronization*, which can be described by phenomenological models such as the Lund string model [7]. Here, the hadronization process is modeled based on the idea of an increasing potential energy stored in strings between two quarks, as schematically illustrated in Fig. 2.2. If the distance between the two quarks increases, it eventually becomes energetically more favorable to produce a new quark-antiquark pair to create new color-neutral bound states.

As further discussed in Chapter 3, Monte Carlo event generators used for simulating collision events model the hadronization process based on such phenomenological models. Experimentally, QCD color confinement and the resulting hadronization of quarks or gluons manifests itself in a collimated stream of hadrons and other particles from their subsequent decays, called a *jet*.

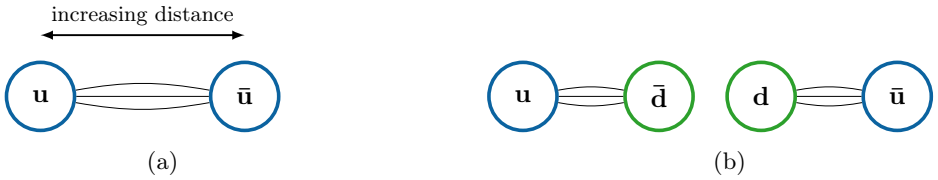


Figure 2.2: Schematic illustration of hadronization in the string model. (a) The potential energy increases with increasing distance. (b) A new quark-antiquark pair is created from the vacuum to form new color-neutral hadrons.

2.1.3 Electroweak Theory

The theory of electroweak interactions by Glashow, Weinberg and Salam [8–10] unifies the electromagnetic and weak interaction based on a $SU_L(2) \times U_Y(1)$ symmetry group. The electroweak theory is a *chiral* theory that distinguishes between the left-handed component ψ_L and right-handed component ψ_R of a fermion field ψ which can be projected out as

$$\psi_L = \frac{1-\gamma^5}{2}\psi, \quad \psi_R = \frac{1+\gamma^5}{2}\psi \tag{2.4}$$

with the Dirac matrix γ^5.
Left-handed fermions form doublets of the *weak isospin* I, the quantum number associated to the $SU_L(2)$ symmetry group, and can be ordered as

$$l_L = \begin{pmatrix} \nu_e \\ e \end{pmatrix}_L, \begin{pmatrix} \nu_\mu \\ \mu \end{pmatrix}_L, \begin{pmatrix} \nu_\tau \\ \tau \end{pmatrix}_L, \tag{2.5}$$

$$q_L = \begin{pmatrix} u \\ d \end{pmatrix}_L, \begin{pmatrix} c \\ s \end{pmatrix}_L, \begin{pmatrix} t \\ b \end{pmatrix}_L. \tag{2.6}$$

While fermion doublets have $I = 1/2$ with its third component $I_3 = +1/2$ and $I_3 = -1/2$ for the upper and lower doublet components respectively, right-handed fermion fields have $I = 0$ and are weak isospin singlets that do not couple to the weak interaction.
A Lagrangian that describes weak interactions of left-handed fermion fields j reads

$$\mathcal{L} = \mathcal{L}_{\text{gauge}} + \mathcal{L}_{\text{fermions}} = -\frac{1}{4}W^{a\mu\nu}W^a_{\mu\nu} + \sum_j \bar{\psi}^j_L \gamma^\mu D_\mu \psi^j_L, \tag{2.7}$$

where invariance under $SU_L(2)$ transformations is ensured by the covariant derivative definition

$$D_\mu = \partial_\mu - i\frac{1}{2}gW^a_\mu \sigma^a \tag{2.8}$$

with coupling constant g, the massless gauge bosons W_μ^a with $a = 1, 2, 3$ and the Pauli matrices σ^a, the generators of $SU_L(2)$. Here, the field strength tensor $W_{\mu\nu}^a$ is defined as

$$W_{\mu\nu}^a = \partial_\mu W_\nu^a - \partial_\nu W_\mu^a + g_w \epsilon^{abc} W_\mu^b W_\nu^c, \tag{2.9}$$

similarly to the QCD field strength tensor giving rise to weak gauge field self-interactions, where ϵ^{abc} denotes structure constants.
When furthermore including the symmetry group of the weak hypercharge $U_Y(1)$ with its corresponding gauge boson field B_μ, the term $\mathcal{L}_{\text{gauge}}$ in Eq. (2.7) becomes

$$\mathcal{L}_{\text{gauge}} = -\frac{1}{4} W^{a\mu\nu} W_{\mu\nu}^a - \frac{1}{4} B^{\mu\nu} B_{\mu\nu} \tag{2.10}$$

with

$$B_{\mu\nu} = \partial_\mu B_\nu - \partial_\nu B_\mu \tag{2.11}$$

while also the covariant derivative retrieves an additional term to ensure gauge invariance:

$$D_\mu = \partial_\mu - i\frac{1}{2} g W_\mu^a \sigma^a - i\frac{1}{2} g' B_\mu \tag{2.12}$$

Here, g' denotes the coupling constant associated to the weak hypercharge Y.
While $SU_L(2)$ only acts on left-handed fermion fields, the $U_Y(1)$ part of the electroweak's symmetry group acts on both left- and right-handed fields which can be seen in the modified $\mathcal{L}_{\text{fermions}}$ term of Eq. (2.7):

$$\begin{aligned} \mathcal{L}_{\text{fermions}} = &\sum_j \bar{\psi}_L^j \gamma^\mu \left(\partial_\mu - i\frac{1}{2} g W_\mu^a \sigma^a - i\frac{1}{2} g' B_\mu \right) \psi_L^j \\ &+ \sum_j \bar{\psi}_R^j \gamma^\mu \left(\partial_\mu - i\frac{1}{2} g' B_\mu \right) \psi_R^j \end{aligned} \tag{2.13}$$

The experimentally observed massive gauge bosons of the weak interaction, $\mathrm{W}^\pm$ and Z^0, are formed by spontaneous symmetry breaking and the Higgs mechanism. This mechanism also allows to generate fermion masses in the electroweak theory which were not included in the above discussed Lagrangians as they would mix left- and right-handed fermion terms and consequently break the gauge invariance.

2.1.4 Spontaneous Symmetry Breaking and the Higgs Mechanism

The concepts of spontaneous symmetry breaking and the Higgs mechanism are best demonstrated by first considering a simplified model with a complex scalar field ϕ and an underlying $U(1)$ gauge symmetry [4]. Its Lagrangian is given by

$$\mathcal{L} = -\frac{1}{4} F^{\mu\nu} F_{\mu\nu} + |D_\mu \phi|^2 - V(\phi), \tag{2.14}$$

with $D_\mu = \partial_\mu + ieA_\mu$, while the scalar potential $V(\phi)$ is chosen as

$$V(\phi) = -\mu^2|\phi|^2 + \frac{\lambda}{2}\left(|\phi|^2\right)^2 \tag{2.15}$$

with $\mu^2 > 0$ and $\lambda > 0$.
The characteristic shape of $V(\phi)$, as illustrated in Fig. 2.3, implies a minimum potential value *not* at $\phi = 0$ but rather at a nonzero value of

$$\langle\phi\rangle = \sqrt{\frac{\mu^2}{\lambda}} \equiv v, \tag{2.16}$$

which is called the *vacuum expectation value* of ϕ. Note that any other point obtained by a $U(1)$ transformation $\phi \to e^{i\alpha}\phi = e^{i\alpha}v$ results in a physically equivalent vacuum state, minimizing the potential $V(\phi)$.
While the underlying Lagrangian fully respects the $U(1)$ gauge symmetry, its nontrivial vacuum state is not invariant under such symmetry transformation. One says that the gauge symmetry is *spontaneously broken* [4].
The field ϕ can be expanded about the vacuum state and expressed in terms of real scalar field fluctuations ϕ_1 and ϕ_2 around v:

$$\phi(x) = v + \frac{1}{\sqrt{2}}\left(\phi_1(x) + i\phi_2(x)\right) \tag{2.17}$$

The potential in Eq. (2.15) can now be rewritten as

$$V(\phi) = -\frac{\mu^4}{2\lambda} + \mu^2\phi_1^2 + \ldots, \tag{2.18}$$

where cubic and quartic terms are omitted for the sake of conciseness. Eq. (2.18) implies that the field ϕ_1 acquired a mass term with mass $m = \sqrt{2}\mu$, while ϕ_2 describes a massless boson. If the field is expanded around $\phi = v$, i.e. $\mathrm{Re}(\phi) = v$ and $\mathrm{Im}(\phi) = 0$, the field excitation along the real axis with a positive second derivative of the potential thus corresponds to a massive particle, while the excitation along the imaginary axis corresponds to a massless one, since the potential is flat in this direction with a vanishing second derivative. The massless boson ϕ_2 is called a *Goldstone boson*, the appearance of which is described by the Nambu-Goldstone theorem [11, 12], stating that every spontaneously broken continuous symmetry generates a massless boson.
The kinetic term of ϕ in Eq. (2.14) can also be rewritten, again omitting cubic and quartic terms for the sake of conciseness, as

$$|D_\mu\phi|^2 = \frac{1}{2}(\partial_\mu\phi_1)^2 + \frac{1}{2}(\partial_\mu\phi_2)^2 + \sqrt{2}evA_\mu\partial^\mu\phi_2 + e^2v^2A_\mu A^\mu + \ldots, \tag{2.19}$$

which leads to a mass term of a gauge boson with mass $m_A^2 = 2e^2v^2$. The additional degree of freedom required for the longitudinal gauge boson polarization is obtained by choosing a gauge such that $\phi_2(x) = 0$ for all x, making the unphysical field ϕ_2 disappear from the theory. The concept of the spontaneous breaking of a continuous gauge symmetry thus

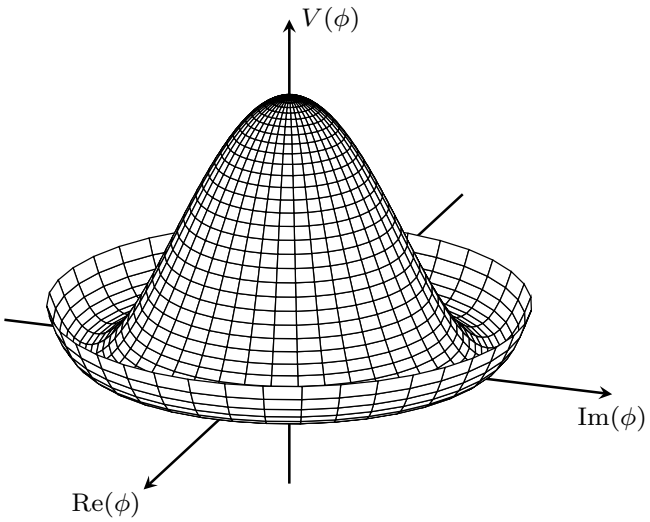


Figure 2.3: Illustration of the potential $V(\phi)$ as defined in Eq. (2.15).

allows to generate a mass term for a gauge boson field A_μ in a gauge-invariant theory and can similarly be applied to the theory of electroweak interactions in the SM, as discussed in the following [4].

The Brout-Englert-Higgs mechanism [13, 14], or short *Higgs mechanism*, adds a new scalar complex field ϕ, the so-called *Higgs field*, to the electroweak theory, which transforms as a doublet of the $SU_L(2)$ group:

$$\phi = \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix} \tag{2.20}$$

The corresponding Lagrangian is given by

$$\mathcal{L}_{\text{Higgs}} = (D_\mu \phi)^\dagger (D^\mu \phi) + \mu^2 \phi^\dagger \phi - \lambda (\phi^\dagger \phi)^2, \tag{2.21}$$

with the covariant derivative D_μ defined in Eq. (2.12). Analogous to the choice of the potential defined in Eq. (2.15), the particular sign choice of the potential terms in $\mathcal{L}_{\text{Higgs}}$ will lead to spontaneous symmetry breaking in the theory with the minimum of the Higgs potential at a vacuum expectation value of the form

$$\langle \phi \rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v \end{pmatrix}, \quad v^2 = \frac{\mu^2}{\lambda}. \tag{2.22}$$

With the vacuum expectation value of ϕ^+ vanishing and ϕ^0 retaining a symmetry under the electromagnetic gauge group $U_{\text{em}}(1)$, the electroweak $SU_L(2) \times U_Y(1)$ symmetry is spontaneously broken down to a $U_{\text{em}}(1)$ symmetry. Thus, the theory will contain one massless gauge boson, corresponding to the unbroken subgroup $U_{\text{em}}(1)$, whereas the remaining three gauge bosons will acquire their masses by the Higgs mechanism and their additional longitudinal polarization degrees of freedom from the three Goldstone bosons originating from the spontaneous breaking of $SU_L(2) \times U_Y(1)$ to $U_{\text{em}}(1)$. The massive Higgs field excitation, corresponding to ϕ_1 in the simplified example above, is called the *Higgs boson*.

Evaluating the kinetic term $(D_\mu\phi)^\dagger(D^\mu\phi)$ in $\mathcal{L}_{\text{Higgs}}$ at the field vacuum expectation value $\langle\phi\rangle$ now leads to gauge boson mass terms in the electroweak theory. Relevant terms are of the form

$$\begin{aligned}\mathcal{L}_{\text{Higgs}} \supset & \frac{1}{2}(0,v)\left(gW_\mu^a\frac{\sigma^a}{2}+\frac{1}{2}g'B_\mu\right)\left(gW^{\mu b}\frac{\sigma^b}{2}+\frac{1}{2}g'B^\mu\right)\begin{pmatrix}0\\v\end{pmatrix}\\ &= \frac{g^2v^2}{8}\left((W_\mu^1)^2+(W_\mu^2)^2\right)+\frac{v^2}{8}(W_\mu^3,B_\mu)\begin{pmatrix}g^2 & -gg'\\ -gg' & g'^2\end{pmatrix}\begin{pmatrix}W^{\mu,3}\\B^\mu\end{pmatrix},\end{aligned} \tag{2.23}$$

from which the three massive gauge bosons of weak interaction

$$\begin{aligned}W_\mu^\pm &= \frac{1}{\sqrt{2}}(W_\mu^1 \mp iW_\mu^2) \quad \text{with mass} \quad m_W = g\frac{v}{2},\\ Z_\mu^0 &= \frac{1}{\sqrt{g^2+g'^2}}(gW_\mu^3-g'B_\mu) \quad \text{with mass} \quad m_Z = \sqrt{g^2+g'^2}\frac{v}{2}\end{aligned} \tag{2.24}$$

can be identified [4].
The fourth vector boson field A_μ, which can be identified as the massless gauge boson of the electromagnetic interaction, remains massless:

$$A_\mu = \frac{1}{\sqrt{g^2+g'^2}}(g'W_\mu^3+gB_\mu) \quad \text{with mass} \quad m_A = 0 \tag{2.25}$$

In addition to generating gauge boson masses for the W and Z bosons while recovering the massless photon of the electromagnetic interaction, the Higgs mechanism also gives masses to the fermion fields. For non-vanishing vacuum expectation values of the Higgs field, Yukawa terms of the form

$$\mathcal{L}_{\text{Y}} \supset -m_f\bar{f}_L f_R + h.c. \tag{2.26}$$

result in non-zero fermion masses

$$m_f = \frac{y_f v}{\sqrt{2}} \tag{2.27}$$

with the dimensionless Yukawa coupling y_f[4].

2.1.5 Limitations of the Standard Model

Despite its remarkable success in explaining a wide range of experimental results and precisely predicting a variety of different physics phenomena and new particles such as the Higgs boson discovered in 2012 by the ATLAS and CMS collaborations at CERN [15, 16], the Standard Model is known to be not complete. Some of its limitations will be briefly discussed in the following.

Dark Matter

Several astrophysical and cosmological observations indicate the existence of a very weakly interacting, non-baryonic and non-luminous form of matter, referred to as *dark matter*,

which accounts for the vast majority of mass in our universe and cannot be explained by the Standard Model.

One of the earliest hints for the existence of dark matter came from galactic rotation curve observations that revealed significant disagreements between the measured circular velocities of luminous objects orbiting the galactic center at large radii and its theoretical predictions based on the visible matter content in the galaxies. According to Newton dynamics, the circular velocity v of an object at large distance r from its galactic center is expected to decrease with increasing distance as $v(r) \propto 1/\sqrt{r}$ [17].

However, several galactic rotation curve measurements, such as those reported in Refs. [18, 19], show an approximately constant evolution for large distances from the galactic center, strongly suggesting the presence of a non-luminous halo of a weakly interacting and thus far unknown type of matter.

Additional evidence for dark matter is provided by gravitational lensing observations of galaxy cluster mergers, such as the so-called *Bullet Cluster* [20]. Here, the collision of two galaxy clusters indicate that the vast majority of the present mass did not undergo interaction during collision, again strongly suggesting the presence of a non-luminous and weakly interacting form of matter, while only the gas content of these clusters (i.e. ordinary, baryonic mass) was affected by the collision and interacted electromagnetically.

Furthermore, the standard cosmological model, or also called ΛCDM (*Lambda Cold Dark Matter*) model, is based on the assumption of a cold (i.e. non-relativistic) dark matter candidate and is able to precisely predict various cosmological observations, such as the observed anisotropies in the cosmic microwave background [21, 22]. Many theories of physics beyond the Standard Model naturally predict so-called *weakly interacting massive particles* (WIMPs) which perfectly fit the model of a relic cold dark matter particle from the early universe. Section 2.2 discusses supersymmetric extensions of the Standard Model, in which WIMPs can appear as the lightest supersymmetric particles and thus form a perfect candidate for dark matter.

Matter-Antimatter Asymmetry

While equal amounts of matter and antimatter should have been produced by the Big Bang, the known universe today almost entirely consists of matter. This observed imbalance between baryonic matter and antibaryonic matter is known as the *baryon asymmetry* or *matter-antimatter asymmetry* problem, as the Standard Model cannot explain this phenomenon. According to the *Sakharov conditions* [23], baryon-generating interactions that are able to produce matter and antimatter at different rates must violate the baryon number as well as the charge (C) and charge-parity (CP) symmetries and additionally take place out of thermal equilibrium. While the Standard Model features CP violation in the quark sector, and thus allows physics to act differently on matter and antimatter, its impact is considered to be too small to account for the observed asymmetry in the universe [24].

Hierarchy Problem

Although the Standard Model is a renormalizable gauge theory, which is in principle well-defined up to infinite energies, it is very well understood that the theory is not complete and new physics will start to appear at some unknown energy scale Λ_{cut} [25]. At the very latest, the current theoretical framework will no longer hold at the Planck scale $M_P \approx 10^{19}\,\text{GeV}$, where quantum gravitation effects start to become import, setting an upper limit on the energy scale Λ_{cut}.

The Higgs boson mass term, composed of its bare mass $m_{h,0}$ and additional radiative corrections arising e.g. from fermion loop diagrams such as the one depicted in Fig. 2.4(a), is given by

$$m_h^2 = m_{h,0}^2 - \frac{|y_f|^2}{8\pi^2}\Lambda_{\text{cut}}^2 + \dots \tag{2.28}$$

and is very sensitive to the energy cut-off scale Λ_{cut} which introduces radiative corrections that are orders of magnitude higher than the experimentally observed Higgs boson mass scale. In order to keep m_h consistent with the experimentally observed value of order $\mathcal{O}(10^2\,\text{GeV})$, an extreme fine-tuning of the involved parameters, leading to an almost exact cancellation between the bare mass and all radiative corrections, is required. This required fine-tuning of mass terms involving energy scales 17 orders of magnitudes higher than the one of the Higgs mass is considered *unnatural* and referred to as the *hierarchy problem*.

One possible solution to the hierarchy problem is offered by supersymmetric extensions of the Standard Model. Here, radiative correction terms involving supersymmetric particles, such as the one shown in Fig. 2.4(b), come with opposite signs compared to their Standard Model partner particles (see Section 2.2) and could therefore lead to a natural cancellation of higher order correction terms [25].

(a) (b)

Figure 2.4: Example radiative corrections to the Higgs mass term involving (a) massive fermions f and (b) a hypothetical massive scalar particle S.

2.2 Supersymmetric Extensions of the Standard Model

Many theories of physics beyond the Standard Model (BSM) have been developed over the past years. Their primary goal is to overcome the limitations seen in the Standard Model while retaining all of the Standard Model properties that led to its tremendous success in predicting and explaining a wide range of physics phenomena. Supersymmetric extensions of the Standard Model are one of the most popular classes of BSM theories as they are often able to offer elegant and natural solutions to many Standard Model shortcomings.

In the following, a brief introduction to supersymmetric extensions of the Standard Model is given.

2.2.1 Supersymmetry Transformations and Supermultiplets

The realization of *supersymmetry* (SUSY) [26–29] in supersymmetric extensions of the Standard Model introduces an additional symmetry to the theory that relates fermions and bosons in such a way that a supersymmetric transformation turns a fermionic state into a bosonic state, and vice versa. An operator generating these transformations is given by an anti-commuting spinor Q with

$$Q\,|\text{Fermion}\rangle = |\text{Boson}\rangle \quad \text{and} \quad Q\,|\text{Boson}\rangle = |\text{Fermion}\rangle\,, \tag{2.29}$$

while also its hermitian conjugate $Q^\dagger$ is a symmetry generator.
As both SUSY generators Q and $Q^\dagger$ commute with the squared-mass operator $-P^2$, SM particles and their respective supersymmetric counterparts, called *superpartners*, have the same eigenvalues of $-P^2$ and therefore the same masses. Furthermore, as Q and $Q^\dagger$ also commute with the generators of SM gauge transformations, SM particles and their respective superpartners are in same representations of the gauge groups and hence must have the same electric charges, weak isospin as well as degrees of freedom in color space. Thus, a SUSY transformation leaves the quantum numbers unchanged except for the spin, which differs by 1/2 between a SM particle and its corresponding superpartner [28, 29].
SM particles and their superpartners are arranged in irreducible representations of the SUSY algebra, called *supermultiplets*, each containing both fermion and boson states and an equal number of fermion and boson degrees of freedom.
Since no supersymmetric partners of SM particles have been observed yet, and in particular none with the exact same masses as their SM partners, SUSY must be – if realized in nature – a broken symmetry. A mechanism similar to the spontaneous symmetry breaking in the electroweak sector could be envisioned for SUSY, such that the theory respects the underlying symmetry but its vacuum state spontaneously breaks SUSY and different masses for particles and their respective superpartners are possible [28, 29].

2.2.2 Minimal Supersymmetric Standard Model

The so-called *Minimal Supersymmetric Standard Model* (MSSM) is the smallest possible supersymmetric extension of the Standard Model as it introduces only a minimal set of additional particle states and interactions. Its particle content and the concept of so-called R-parity conservation will be briefly discussed in the following [28–30].

MSSM Particle Content

In the MSSM, each known SM particle is arranged in a supermultiplet with one respective superpartner particle, also referred to as *sparticle*, as shown in Table 2.1 [28].
The spin-0 superpartners of the SM leptons and quarks are referred to as *sleptons* and

Names (SM, SUSY)	SM field	Superpartner	$SU_C(3)$	$SU_L(2)$	$U_Y(1)$
Quarks, Squarks	$(u_L\ d_L)$	$(\tilde{u}_L\ \tilde{d}_L)$	3	2	$\frac{1}{6}$
($\times$3 families)	$u_R^\dagger$	$\tilde{u}_R^*$	3	1	$-\frac{2}{3}$
	$d_R^\dagger$	$\tilde{d}_R^*$	3	1	$\frac{1}{3}$
Leptons, Sleptons	$(\nu\ e_L)$	$(\tilde{\nu}\ \tilde{e}_L)$	1	2	$-\frac{1}{2}$
($\times$3 families)	$e_R^\dagger$	$\tilde{e}_R^*$	1	1	1
Gluons, Gluinos	g	$\tilde{g}$	8	1	0
W, Winos	$W^a(a=1,2,3)$	$\tilde{W}^a(a=1,2,3)$	1	3	0
B, Bino	B^0	$\tilde{B}^0$	1	1	0
Higgs, Higgsinos	$(H_u^+\ H_u^0)$	$(\tilde{H}_u^+\ \tilde{H}_u^0)$	1	2	$\frac{1}{2}$
	$(H_d^0\ H_d^-)$	$(\tilde{H}_d^0\ \tilde{H}_d^-)$	1	2	$-\frac{1}{2}$

Table 2.1: Particle content of the MSSM. Adapted from Ref. [28].

squarks, respectively, and with left- and right-handed fermions transforming differently in the SM, each of them is assigned a respective supersymmetric counterpart. The naming convention for all sleptons and squarks adds the prefix "s" to every SM particle name, such that, for instance, the muon superpartner is called *smuon* and the top-quark superpartner *stop*.

The SM gauge bosons, on the other hand, are partnered with spin-1/2 fermion fields, referred to as *gauginos*, where the suffix "ino" is used as a naming convention (e.g. *Wino* or *gluino*). Lastly, in order to avoid gauge anomalies in the electroweak sector and allow for a mechanism to give masses to the quarks via Yukawa couplings, the SM Higgs boson in the MSSM is given by a linear combination of the neutral components of the two doublet complex scalar fields H_u and H_d, which are partnered with their respective supersymmetric counterparts, called *Higgsinos* [30].

Similar to, for instance, the mixing of the gauge eigenstates W_μ^a and B_μ to mass eigenstates $W_\mu^\pm$, Z_μ^0 and A_μ in the SM electroweak sector, the supersymmetric gauge eigenstates can mix to form different mass eigenstates after electroweak and supersymmetry breaking, as summarized in Table 2.2. While the winos and charged higgsinos mix to form the charged mass eigenstates $\tilde{\chi}_1^\pm$ and $\tilde{\chi}_2^\pm$, referred to as *charginos*, the gauginos and neutral higgsinos combine to form the four neutral mass eigenstates $\tilde{\chi}_1^0$, $\tilde{\chi}_2^0$, $\tilde{\chi}_3^0$ and $\tilde{\chi}_4^0$, the *neutralinos*. By convention, mass eigenstates are always labeled in ascending mass order, such that the lightest neutralino is represented by $\tilde{\chi}_1^0$.

In the squark and slepton sector, *Minimal Flavor Violation* (MFV) [31] dictates flavor mixing to be linked to the known structure of Yukawa couplings, motivating the assumption of nearly mass-degenerate, unmixed states in the first two squark and slepton generations due to their negligibly small Yukawa couplings. Gauge eigenstates of the third generations, on the other hand, mix to form the stop and sbottom mass eigenstates $\tilde{t}_1$, $\tilde{t}_2$, $\tilde{b}_1$ and $\tilde{b}_2$ in

the squark sector as well as $\tilde{\tau}_1$, $\tilde{\tau}_2$ and $\tilde{\nu}_\tau$ in the slepton sector. However, non-MFV models that allow for a significant mixing between the second and third squark generations are also well motivated (see e.g. Refs. [32, 33]) and play an important role in the context of this thesis, as further discussed in Section 2.2.3.

Lastly, the Higgs boson mass eigenstates h^0, H^0, A^0 and $H^\pm$ are formed from the mixing of the Higgs gauge eigenstates. Here, h^0 can be identified with the known SM Higgs boson, while the additional Higgs boson H^0, A^0 and $H^\pm$ are heavier, so far unobserved particles. Together with the mixing angles and unknown masses of these newly predicted particles as free parameters of the theory, the MSSM adds a total of 105 additional parameters to the already 19 existing free parameters of the SM. Although phenomenological constraints can be used to reduce the 105 parameters to only 19 additional parameters in the so-called *phenomenological Minimal Supersymmetric Standard Model* (pMSSM) [34], the resulting parameter space is still extremely complex and would lead to unmanageable computational complexity when being probed in SUSY searches at collider experiments. In order to reduce the parameter space to be covered in a data analysis to more manageable levels, simplified SUSY models are commonly used in collider searches. Only a few new SUSY particles are considered in these simplified models, with their masses and production cross-sections as free parameters, while all other sparticles are assumed to be kinematically inaccessible at the current collider energies. The SUSY search presented in Chapter 9, for instance, is based on a simplified model, considering only the lightest supersymmetric top $\tilde{t}_1$ as well as the lightest neutralino $\tilde{\chi}_1^0$ to be experimentally accessible.

Name	Spin	Gauge Eigenstate	Mass Eigenstate
Higgs bosons	0	$H_u^0\ H_d^0\ H_u^+\ H_d^-$	$h^0\ H^0\ A^0\ H^\pm$
Squarks	0	$\tilde{u}_L\ \tilde{u}_R\ \tilde{d}_L\ \tilde{d}_R$	(same)
		$\tilde{s}_L\ \tilde{s}_R\ \tilde{c}_L\ \tilde{c}_R$	(same)
		$\tilde{t}_L\ \tilde{t}_R\ \tilde{b}_L\ \tilde{b}_R$	$\tilde{t}_1\ \tilde{t}_2\ \tilde{b}_1\ \tilde{b}_2$
Sleptons	0	$\tilde{e}_L\ \tilde{e}_R\ \tilde{\nu}_e$	(same)
		$\tilde{\mu}_L\ \tilde{\mu}_R\ \tilde{\nu}_\mu$	(same)
		$\tilde{\tau}_L\ \tilde{\tau}_R\ \tilde{\nu}_\tau$	$\tilde{\tau}_1\ \tilde{\tau}_2\ \tilde{\nu}_\tau$
Neutralinos	$\frac{1}{2}$	$\tilde{B}^0\ \tilde{W}^0\ \tilde{H}_u^0\ \tilde{H}_d^0$	$\tilde{\chi}_1^0\ \tilde{\chi}_2^0\ \tilde{\chi}_3^0\ \tilde{\chi}_4^0$
Charginos	$\frac{1}{2}$	$\tilde{W}^\pm\ \tilde{H}_u^+\ \tilde{H}_d^-$	$\tilde{\chi}_1^\pm\ \tilde{\chi}_2^\pm$
Gluinos	$\frac{1}{2}$	$\tilde{g}$	(same)

Table 2.2: Gauge and mass eigenstates of the (undiscovered) particles in the MSSM, where mixing in the first and second sfermion generation is assumed to be negligible. Adapted from Ref. [28].

R-Parity

The MSSM introduces new possible interactions that allow for lepton number (L) and baryon number (B) violating processes which would lead to the decay of the proton, as illustrated in Fig. 2.5. Since both L and B violating processes have never been observed experimentally, with the strongest constraints set by proton lifetime measurements with current limits of more than $\sim 10^{33}$ years [35–37], a new symmetry, called R-parity [38], is introduced to the MSSM. R-parity is a multiplicatively conserved quantum number, defined for each particle as

$$P_R = (-1)^{3(B-L)+2s}, \tag{2.30}$$

where its spin is denoted by s.
Given the definition in Eq. (2.30), all SM particles as well as all Higgs boson have even R-parity with $P_R = +1$, whereas all SUSY particles come with $P_R = -1$.
In addition to fixing the proton lifetime, the definition of R-parity conservation leads to the following very important phenomenological consequences:

- Supersymmetric interaction vertices always contain an even number of SUSY particles. Sparticles are therefore always produced in pairs and can only decay into an odd number of other sparticles.
- Heavy SUSY particle will always undergo cascade decays into final states involving the *lightest supersymmetric particle* (LSP) which must be stable and cannot further decay.

An electrically neutral LSP would not only be stable due to R-parity conservation, but would additionally interact only weakly with ordinary matter, making it a perfect WIMP candidate to explain the elusive nature of non-baryonic dark matter [28].

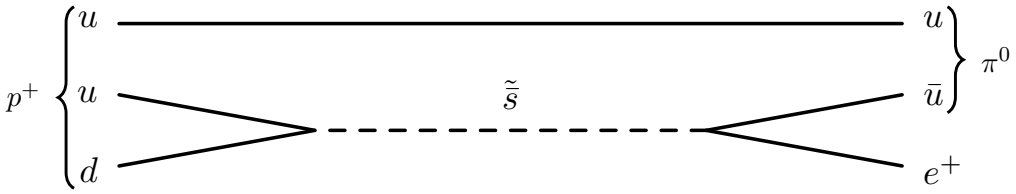


Figure 2.5: Exemplary proton decay diagram in case of R-parity violation in a supersymmetric SM extension.

2.2.3 Flavor-Violation in Supersymmetric Models

While most SUSY searches are based on simplified models that consider new supersymmetric particles that undergo a single well-defined decay mode, the structure of the MSSM could be more complex and result in decay topologies that are not covered by current searches. The effects of squark mixing in the second and third generation, in particular $\tilde{c}_{L/R} - \tilde{t}_{L/R}$ mixing, resulting in mass eigenstates that contain multiple well-defined flavor components and therefore allow for multiple different decay modes, was already investigated in 2011 in Ref. [33]. A dedicated search for final states containing a top quark, a

charm quark and missing transverse momentum was proposed which could significantly enhance the experimental sensitivity in case that squarks are not flavor eigenstates but instead feature a sizable flavor mixing.

A reinterpretation of an ATLAS 36.1 fb^{-1} search for top pair production with one lepton, jets and missing transverse momentum ($E_{\text{T}}^{\text{miss}}$) [39] as well as an ATLAS 36.1 fb^{-1} search for charm pair production with $E_{\text{T}}^{\text{miss}}$ [40] was performed in Ref. [41]. The reinterpretation considers a simplified model with two active squark flavors, a right-handed charm squark $\tilde{c}_R$ and a right-handed top squark $\tilde{t}_R$, which mix to two physical mass eigenstates $\tilde{t}_1$ and $\tilde{t}_2$ with its flavor structure governed by the θ_{tc} mixing angle according to

$$\begin{pmatrix} \tilde{t}_1 \\ \tilde{t}_2 \end{pmatrix} = \begin{pmatrix} \cos\theta_{tc} & \sin\theta_{tc} \\ -\sin\theta_{tc} & \cos\theta_{tc} \end{pmatrix} \begin{pmatrix} \tilde{c}_R \\ \tilde{t}_R \end{pmatrix}. \tag{2.31}$$

The lighter sparticle $\tilde{t}_1$ is assumed to be kinematically accessible and decays into $\text{t}/\bar{\text{t}} + \tilde{\chi}_1^0$ or $\text{c}/\bar{\text{c}} + \tilde{\chi}_1^0$, where the neutralino $\tilde{\chi}_1^0$ is the LSP and therefore a dark matter candidate. Fig. 2.6 shows the resulting exclusion limits, where for large mixings of $\theta_{tc} \simeq \frac{\pi}{4}$ a significant reduction in sensitivity to less than 600 GeV in $\tilde{t}_1$ mass is seen for the existing searches. A dedicated search for a $\text{t}\bar{\text{c}}/\text{c}\bar{\text{t}} + E_{\text{T}}^{\text{miss}}$ (from now on referred to as $\text{tc} + E_{\text{T}}^{\text{miss}}$) final state resulting from non-MFV SUSY was investigated and found to be ideal to target this largely unexplored phase space.

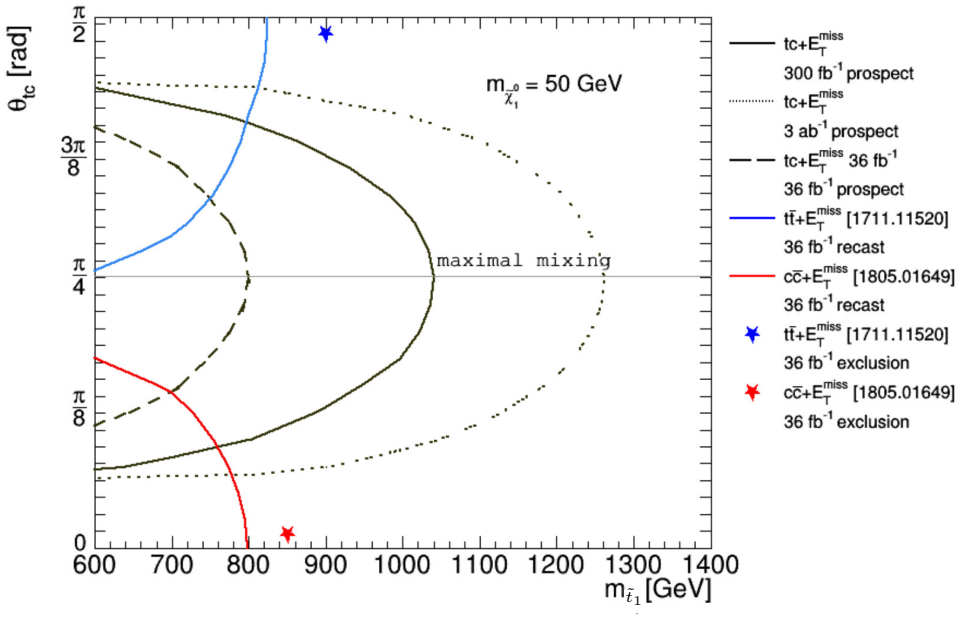


Figure 2.6: Exclusion limits for a simplified $\text{tc} + E_{\text{T}}^{\text{miss}}$ model based on the reinterpretations of ATLAS 36.1 fb^{-1} $\text{t}\bar{\text{t}} + E_{\text{T}}^{\text{miss}}$ (blue) and $\text{c}\bar{\text{c}} + E_{\text{T}}^{\text{miss}}$ (red) searches. Expected exclusion limits for a dedicated $\text{tc} + E_{\text{T}}^{\text{miss}}$ search (black) are also shown, covering a large proportion of the unexplored phase space. Figure adapted from Ref. [41].

As shown in Fig. 2.7, considering a maximal mixing of $\theta_{tc} = \frac{\pi}{4}$ leads to branching ratios (BRs) for the decays $\tilde{t}_1 \to c\tilde{\chi}_1^0$ and $\tilde{t}_1 \to t\tilde{\chi}_1^0$ of

$$\begin{aligned} \mathrm{BR}(\tilde{t}_1 \to c\tilde{\chi}_1^0) &\simeq 0.5, \\ \mathrm{BR}(\tilde{t}_1 \to t\tilde{\chi}_1^0) &\simeq 0.5 \end{aligned} \tag{2.32}$$

and thus, the three different signal topologies of pair-produced $\tilde{t}_1$ decays depicted in Fig. 2.8 occur with relative rates of 50% for $t\bar{c}/\bar{t}c+\tilde{\chi}_1^0+\tilde{\chi}_1^0$ final states as well as 25% for $t\bar{t}+\tilde{\chi}_1^0+\tilde{\chi}_1^0$ and $c\bar{c}+\tilde{\chi}_1^0+\tilde{\chi}_1^0$, respectively.
Following this approach for a proposed dedicated $tc + E_{\mathrm{T}}^{\mathrm{miss}}$ analysis, an ATLAS search for non-MFV SUSY assuming a maximal mixing of $\theta_{tc} = \frac{\pi}{4}$ was conducted and will be presented in Chapter 9 of this thesis.

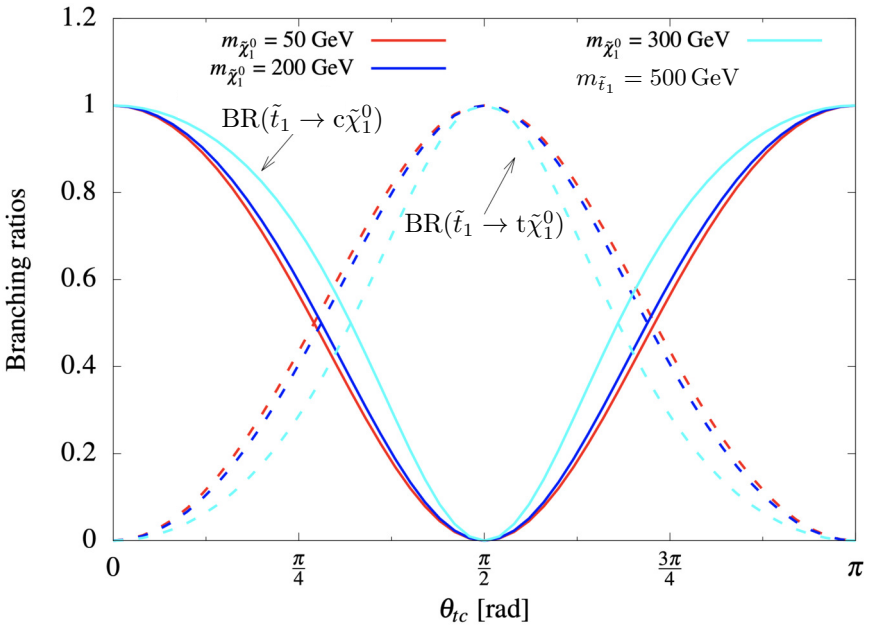
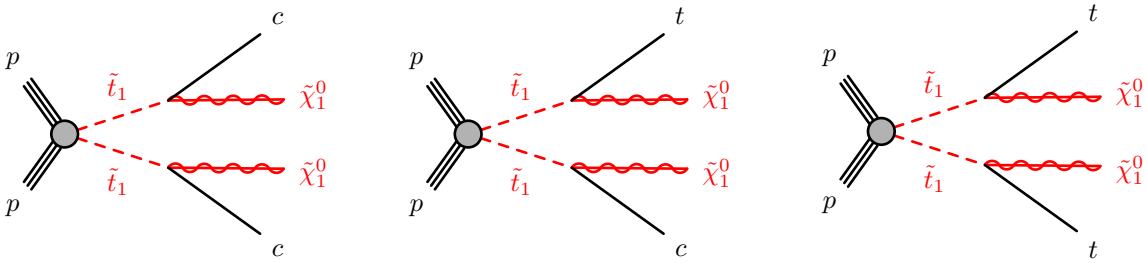


Figure 2.7: Dependence of the branching ratios $\mathrm{BR}(\tilde{t}_1 \to c\tilde{\chi}_1^0)$ (solid) and $\mathrm{BR}(\tilde{t}_1 \to t\tilde{\chi}_1^0)$ (dashed) on the mixing angle θ_{tc} for different neutralino masses and a fixed stop mass $m_{\tilde{t}_1} = 500\,\mathrm{GeV}$. Figure adapted from Ref. [41].



(a) $pp \to c\bar{c} + \tilde{\chi}_1^0 + \tilde{\chi}_1^0$ (b) $pp \to t\bar{c}/\bar{t}c + \tilde{\chi}_1^0 + \tilde{\chi}_1^0$ (c) $pp \to t\bar{t} + \tilde{\chi}_1^0 + \tilde{\chi}_1^0$

Figure 2.8: Diagrammatic representations of the decay topologies of pair-produced $\tilde{t}_1$ squarks in proton-proton (pp) collision, assuming a non-negligible flavor mixing in the second and third squark generations.

Chapter 3

The ATLAS Experiment at the Large Hadron Collider

3.1 The Large Hadron Collider

The Large Hadron Collider [42] is currently the world's largest and most powerful particle accelerator, located at the European Organization for Nuclear Research (CERN) near Geneva, Switzerland. Two counter-rotating beams of protons — and in dedicated runs also beams of other hadrons such as Pb-ions — are accelerated to almost the speed of light along the 26.7 km ring and brought into collision in four interaction regions in which large experiments are situated to detect, reconstruct and analyze the collision events.
Important machine aspects and parameters are briefly discussed in the following.

3.1.1 Accelerator Chain and Experiments at the LHC

As shown in Fig. 3.1, the LHC is part of CERN's accelerator complex and the last stage in a chain of smaller accelerators that are used to gradually boost the particles to their final energies. After being produced by ionizing hydrogen atoms, protons start their journey in the LHC injection chain by entering a linear accelerator: for the last few decades, and thus also during LHC Run 1 (2010-2013) and Run 2 (2015-2018), LINAC2 [43] served as the first link in CERN's accelerator chain and was recently replaced by CERN's currently newest accelerator, LINAC4 [44]. Here, protons are accelerated to an energy of 160 MeV before being injected into three successive synchrotron systems, the Proton Synchrotron Booster (BOOSTER) [45], the Proton Synchrotron (PS) [46] and the Super Proton Synchrotron (SPS) [47], which accelerate the particles to progressively higher energies of 1.4 GeV, 25 GeV and 450 GeV, respectively.
Finally, the protons are injected into the LHC as two beams, traveling in two separate beam pipes in opposite directions.
Pb-ions for the LHC, on the other hand, start from a source of vaporized lead, entering the accelerator chain through LINAC3 [48] before being accelerated in the Low Energy Ion Ring (LEIR) [49], after which the particles follow the same route to maximum energy as the protons described above.
The LHC comprises 1232 superconducting dipole magnets, each 15 m in length and oper-

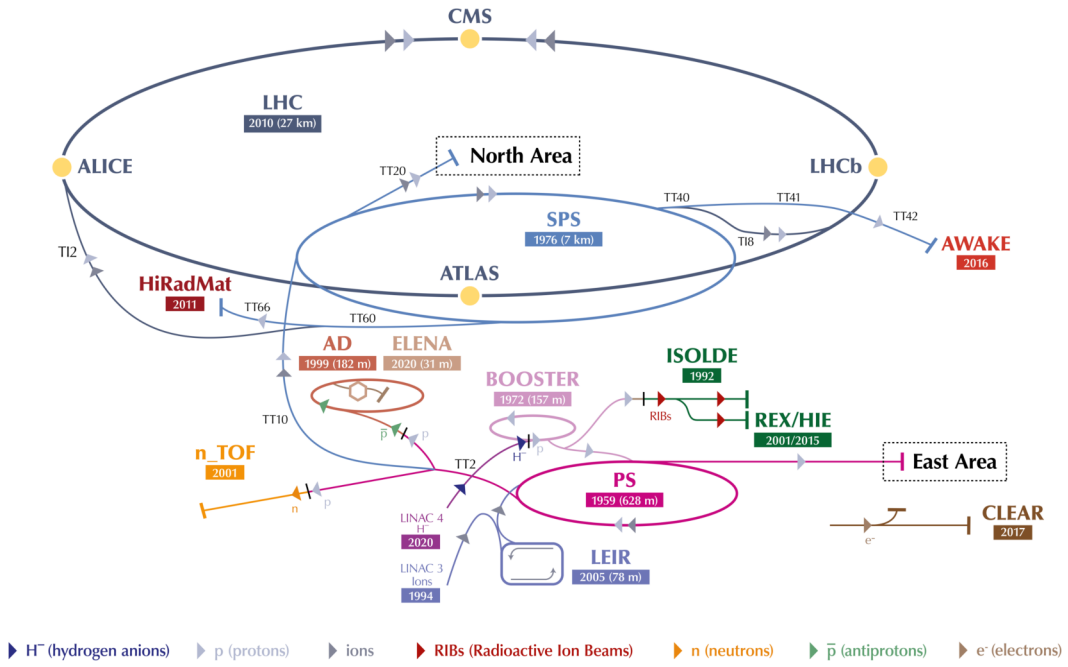


Figure 3.1: The CERN accelerator complex. A chain of accelerators is used to provide particle beams to several different experiments as well as to boost the particles to their final energies in the last stage of the CERN accelerator complex, the LHC [50].

ated at a temperature of 1.9 K, providing the necessary transverse acceleration needed to keep the charged particles on their circular trajectory along the accelerator ring. Superconducting radio-frequency cavities accelerate the particles in the longitudinal direction to a final proton-proton (pp) collision energy of 6.5 TeV, resulting in a center-of-mass energy $\sqrt{s} = 13\,\text{TeV}$, which is close to the LHC design value for pp collisions of 14 TeV. Each of the two particle beams nominally consists of 2808 bunches of about 10^{11} protons per bunch, which are separated by 25 ns in time and therefore result in a bunch-crossing frequency of 40 MHz in the four main LHC experiments ALICE, ATLAS, CMS and LHCb. A single bunch crossing is usually referred to as an *event*.

Each of the four experiments is housed in a huge underground cavern along the particle collider and is specifically designed to measure and analyze the collision events provided by the LHC. The detector geometries as well as the sub-detector components are highly specialized to the physics goals of the respective experiments, which will be briefly discussed in the following:

- **ALICE** [51] is a dedicated heavy-ion particle detector with its main goal to study strongly interacting matter at extremely high energy densities and temperatures, where quark-gluon plasma is expected to exist.
- **ATLAS** [52] is the world's largest-volume particle detector at a particle accelerator and is one of the two large multi-purpose detectors at the LHC. It covers a broad range of particle physics research and is specifically designed to fully exploit the discovery potential of the LHC. ATLAS is optimized for the discovery of the Higgs boson and precise measurements of its properties and complements its physics program with

SM precision tests, searches beyond the SM as well as heavy-ion physics.

- **CMS** [53] is the second large multi-purpose detector at the LHC, following a physics program similar to the one of ATLAS.
- **LHCb** [54] is a forward spectrometer, specialized on precision studies of CP violation and b-physics and therefore specifically aiming to provide insights into phenomena like matter-antimatter asymmetry.

3.1.2 Luminosity and Pile-up

Apart from the center-of-mass energy a particle collider can reach, another key parameter of interest is the *instantaneous luminosity* $\mathcal{L}$, a quantity that measures a collider's ability to produce interactions of a given process P at a rate

$$\frac{\mathrm{d}N_P}{\mathrm{d}t} = \sigma_P \times \mathcal{L}, \tag{3.1}$$

where σ_P corresponds to the respective interaction cross-section [55].
Usually high values of instantaneous luminosity are desired in order to be able to study rare processes with very small interaction cross-sections. While Eq. (3.1) serves to understand the importance of instantaneous luminosity in statistical analyses, luminosity can be seen as a property of a particle collider and its beam structure. For two head-on colliding beams in a circular accelerator with identical Gaussian beam profiles, the instantaneous luminosity can be expressed as

$$\mathcal{L} = \frac{n_p^2 n_b f}{4\pi\sigma_x\sigma_y} \cdot S, \tag{3.2}$$

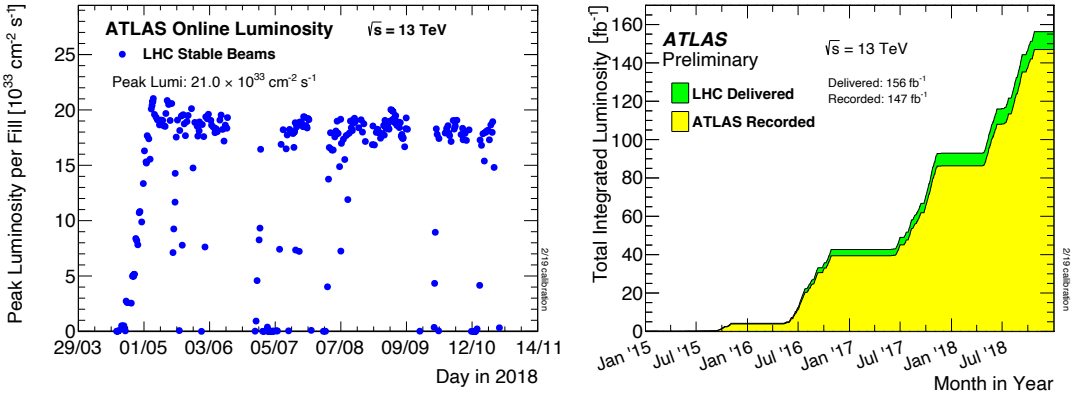
where n_p is the number of protons per bunch, n_b the total number of bunches per beam, f the bunch revolution frequency, σ_x and σ_y the beam resolution in the transverse plane at the collision point and S the so-called luminosity reduction factor, accounting for non-zero crossing angles of the particle bunches [55].
The LHC has a design instantaneous luminosity of $\mathcal{L} = 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, a value that was already well exceeded in the past, as can be seen in Fig. 3.2(a).
The total number of expected interactions of a given process P in a time period T is now given by

$$N_P = \int_0^T \frac{\mathrm{d}N_P}{\mathrm{d}t}\mathrm{d}t = \sigma_P \times \int_0^T \mathcal{L}(t)\mathrm{d}t = \sigma_P \times \mathcal{L}_{\mathrm{int}}, \tag{3.3}$$

where

$$\mathcal{L}_{\mathrm{int}} = \int_0^T \mathcal{L}(t)\mathrm{d}t \tag{3.4}$$

denotes the so-called *integrated luminosity*, a quantity that is often used to quantify the amount of data an experiment has recorded and is typically measured in inverse femtobarn (fb^{-1}). Fig. 3.2(b) shows the LHC delivered and ATLAS recorded cumulative integrated luminosity during stable beams for pp collisions at 13 TeV center-of-mass energy in LHC Run 2.



(a) (b)

Figure 3.2: (a) LHC peak instantaneous luminosity per fill delivered to ATLAS and (b) LHC delivered as well as ATLAS recorded total integrated luminosity during stable beams for pp collisions at $\sqrt{s} = 13\,\text{TeV}$ in LHC Run 2 [56].

While high instantaneous luminosities are evidently beneficial for probing low cross-section processes and therefore highly desired, it leads to the occurrence of a non-negligible number of simultaneous pp interactions during a single bunch crossing, referred to as *pile-up* (PU). These additional collisions are superimposed to the interesting physics process that usually causes an event to be stored for further analysis (see Section 3.2.5), the so-called *hard-scatter* (HS) interaction, and therefore require experiments to implement sophisticated techniques to distinguish between the different interaction types and their associated particles in the detector.

In fact, a large part of this thesis deals with the implementation and development of efficient algorithms for the reconstruction of the primary pp interaction vertices, whose knowledge is essential for understanding and mitigating pile-up effects.

The expected number of pile-up collisions per bunch crossing μ is given by

$$\mu = \frac{\sigma_{\text{inelastic}} \mathcal{L}}{n_b f}, \tag{3.5}$$

where $\sigma_{\text{inelastic}}$ denotes the inelastic pp interaction cross-section. Fig. 3.3 shows the luminosity-weighted distribution of the mean number of interactions per bunch crossing for LHC Run 2 data taking with a total average of $\langle\mu\rangle = 33.7$.

In additional to the above described *in-time* pile-up of overlaying pp collisions during a single bunch crossing, interactions from adjacent bunch crossings, referred to as *out-of-time* pile-up, can also contaminate signals in detector systems with integration times that are significantly larger than the 25 ns bunch spacing. This is for instance the case in the ATLAS electromagnetic calorimeter system (see Section 3.2.3) and dedicated correction techniques are usually applied to estimate and mitigate these effects [57].

3.1.3 The High-Luminosity Upgrade

The High-Luminosity LHC (HL-LHC) [58] will be a major upgrade of the Large Hadron Collider, aiming to sustain or even further extend its discovery potential. The upgrade

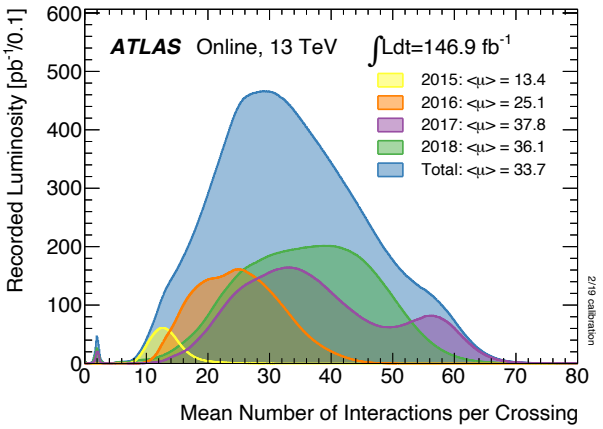


Figure 3.3: Luminosity-weighted distribution of the mean number of interactions per bunch crossing in ATLAS for each year of Run 2 pp collisions at 13 TeV center-of-mass energy. [56]

will increase the collider's instantaneous luminosity by more than a factor of five beyond its original design value and thus also the total expected integrated luminosity by a factor of ten to $\sim 3000\,\mathrm{fb}^{-1}$ after about tens years of operation.

A number of key innovative technologies will be essential for the novel HL-LHC machine configuration, such as cutting-edge 11 T superconducting magnets, very compact superconducting so-called crab cavities for beam rotation or new technologies for beam collimation. The resulting increased instantaneous luminosity of up to $\mathcal{L} = 5 - 7.5 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ will accordingly increase the average number of simultaneous pp collisions during a single bunch crossing in the detectors to up to $\langle\mu\rangle = 200$ and will therefore pose unprecedented challenges to all sub-detector systems as well as to all event processing software components. Section 3.3 will briefly present some of the detector upgrades planned for the ATLAS experiment in preparation for the HL-LHC era.

3.2 The ATLAS Detector

ATLAS is a forward-backward symmetric cylindrical detector with a close to 4π coverage of solid angle, measuring 46 m in length and 25 m in diameter with a weight of around 7000 t. As shown in Fig. 3.4, it is comprised of various sub-detector systems that are arranged in a structure of cylindrical layers, each of them being specialized on the precise measurement of specific properties of the particles emerging from the interaction point in the center of the detector. A tracking system, responsible for the reconstruction of charged particle trajectories and interaction vertices, surrounding the nominal collision point, forms the innermost detector system of ATLAS. It is surrounded by a solenoid magnet system, followed by an electromagnetic and hadronic calorimeter for energy measurements. The detector is complemented by a muon tracking system, the outermost part of ATLAS, which is immersed in a magnetic field provided by the toroid magnets.
A more detailed overview of the various detector components and the ATLAS coordinate system, largely based on Ref. [52], will be given in the following.

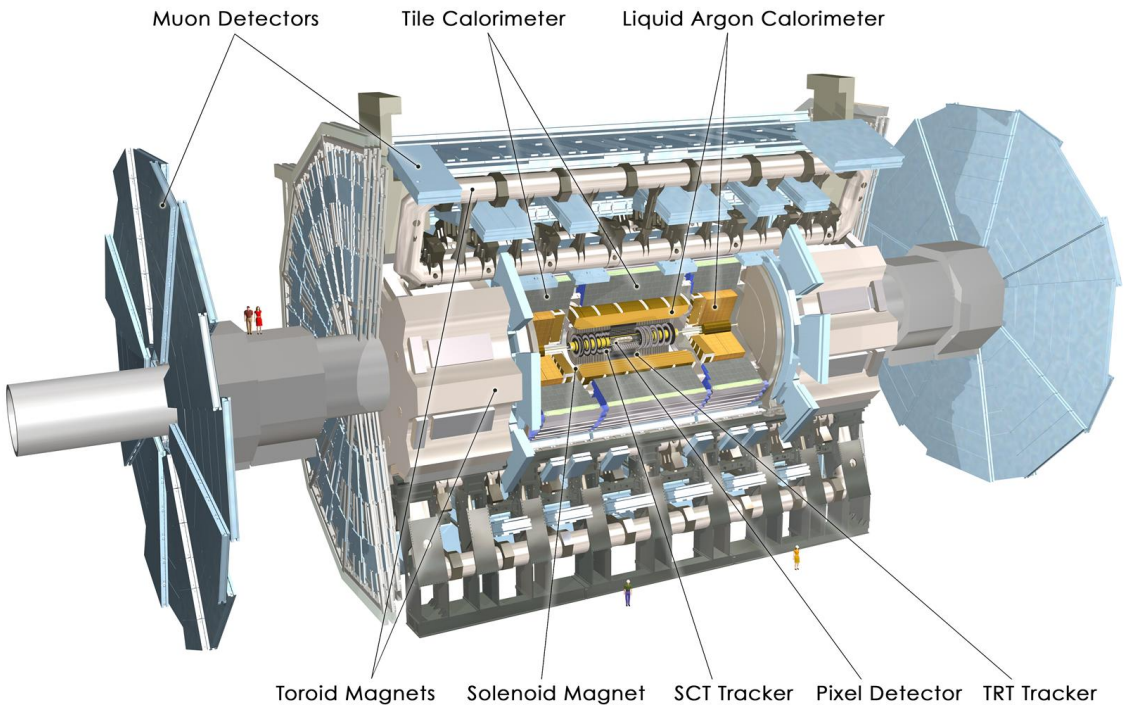


Figure 3.4: Schematic illustration of the ATLAS detector at the LHC at CERN with its sub-detector and magnet systems [59].

3.2.1 Coordinate System and Definitions

The ATLAS coordinate system is defined as a right-handed coordinate system with its origin in the nominal collision point in the center of the detector. As depicted in Fig. 3.5, the z-axis is chosen to coincide with the LHC beam axis, the positive x-axis points towards the center of the LHC ring and the positive y-axis upwards to the surface of the Earth. The x-y plane is referred to as the *transverse plane*, in which the azimuthal angle ϕ is

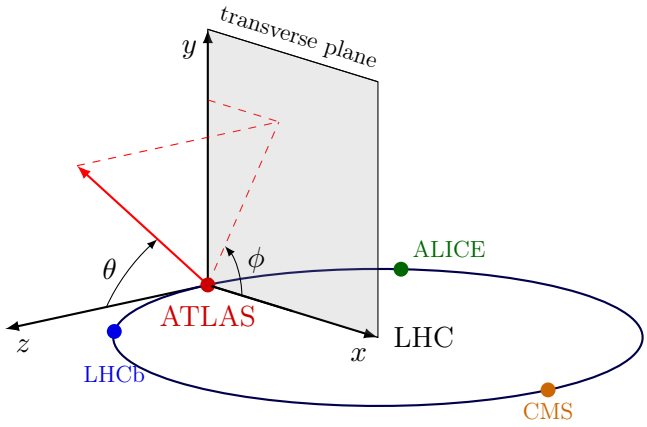


Figure 3.5: The ATLAS coordinate system.

defined relative to the x-axis. Additionally, the polar angle θ is defined as the angle from the positive z-axis in the y-z plane.
Instead of the polar angle θ, a massless approximation of the Lorentz-invariant rapidity, the so-called *pseudo-rapidity* η, is often used. It is defined as

$$\eta = \ln \frac{|\vec{p}| + p_L}{|\vec{p}| - p_L} = -\ln \tan \frac{\theta}{2}, \tag{3.6}$$

where $\vec{p}$ and p_L denote the momentum as well as the longitudinal momentum of a particle, respectively.
Using the definition of Eq. (3.6), distances between physics objects in the detector can be quantified with the distance measure

$$\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2}. \tag{3.7}$$

The transverse momentum $p_T = \sqrt{p_x^2 + p_y^2}$ as well as the transverse energy $E_T = \sqrt{m^2 + p_T^2}$ for a particle with mass m are important quantities as they allow to identify particles that have left the detector without being detected: since the colliding partons have close-to-zero and thus negligible momentum in the transverse plane in their initial state, the total transverse momentum sum of all produced particles in the final state has to add up to zero as well by virtue of momentum conservation. Any value significantly different from zero implies particles that have left the detector undetected.

3.2.2 Inner Detector and Central Solenoid

The Inner Detector (ID) forms the innermost part of the ATLAS detector and is dedicated to the precise reconstruction of charged particle trajectories as well as positions of interaction vertices. The 7 m long cylindrical detector system is fully immersed in a 2 T solenoidal magnetic field, provided by the ATLAS central solenoid magnet system, which allows to extract charge and momentum information of the traversing particles. While the ATLAS central solenoid is able to produce a highly homogeneous magnetic field along the

z-axis in its center, inhomogeneities are present at its boundaries.
As schematically shown in Fig. 3.6, the Inner Detector starts at a radius of 33.25 mm from the beam axis and extends to a radius of 1.082 m, consisting of three different sub-detector systems, a silicon pixel detector (Pixels), a semiconductor tracker (SCT) and a transition radiation tracker (TRT). Each subsystem comprises a barrel and two end-cap sections (see Fig. 3.6(a) and Fig. 3.6(b), respectively), providing a total coverage of 2π in azimuthal angle ϕ as well as a pseudo-rapidity coverage of up to $|\eta| < 2.5$. The Insertable B-Layer (IBL) was added to ATLAS during the long shutdown between LHC Run 1 and Run 2 and is part of the Pixel barrel section.
A more detailed description of the various ID sub-detector components will be given in the following.

Pixel Detector

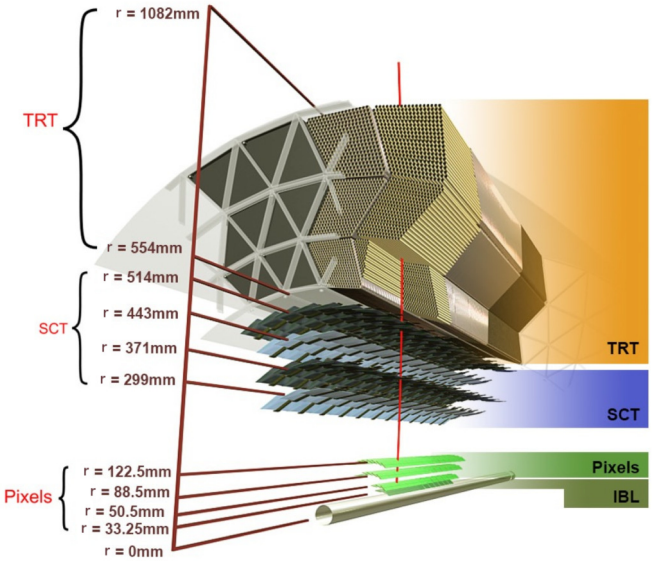
The pixel detector [60] is the innermost element of the Inner Detector and is able to provide extremely high resolution measurements that play a crucial role in the precise reconstruction of tracks, primary interaction vertices and the identification of long-lived particles, such as B-hadrons. It is composed of three concentric barrel layers, positioned at radii of 50.5 mm, 88.5 mm and 122.5 mm as well as two identical end-cap regions, each comprising three pixel disks at $|z| = 495$ mm, 580 mm, 650 mm (see Fig. 3.7). The detector layout provides a full tracking coverage of $|\eta| \leq 2.5$ for the barrel section and $1.7 \leq |\eta| \leq 2.5$ for the end-caps.
Additionally, the Insertable B-Layer [61] is positioned closest to the interaction region at a radius of 33.5 mm, significantly improving the pixel detector's performance in more dense environments.
The pixel detector consists of 1744 silicon sensor modules, each 19 mm × 63 mm in size, 250 μm in depth and comprising 47 232 individual pixels. The nominal pixel size is 50 μm × 400 μm (about 90% of all pixels) while a small fraction (about 10% of all pixels) feature a size of 50 μm × 600 μm, resulting in a total area of about 1.7 m^2 with more than 80 million read-out channels. The IBL adds another 13 178 880 individual silicon pixels of size 50 μm × 250 μm, arranged in 286 modules.
A traversing charged particle will create electron-hole pairs in these silicon detector layers, which will then, by virtue of an applied bias voltage, drift to the anode and cathode part of the pixel sensors and can subsequently be read out by dedicated electronics.

Semiconductor Tracker

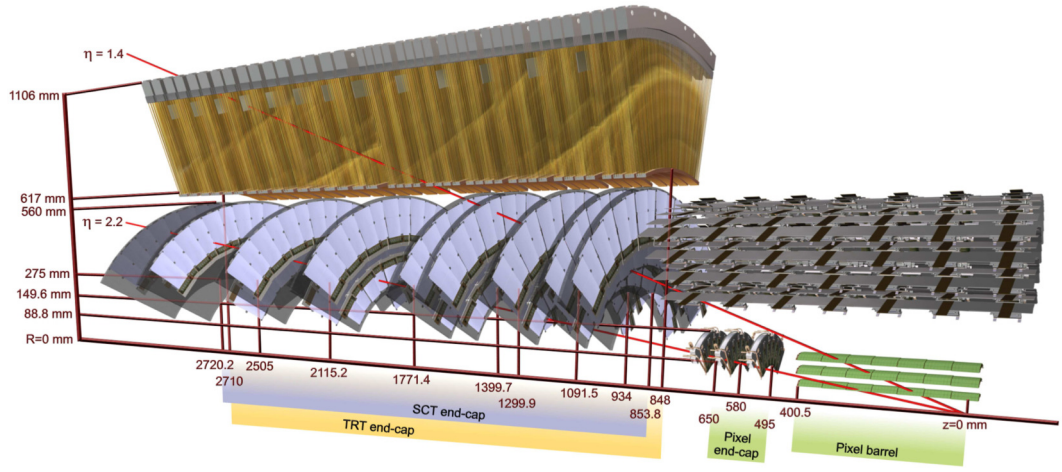
Surrounding the pixel detector, the barrel part of the Semiconductor Tracker (SCT) [62] is located, comprising four cylindrical silicon detector layers at radii of 299 mm, 371 mm, 443 mm and 514 mm with a total of 2112 modules. As shown in Fig. 3.7, the SCT is complemented by an end-cap system on both sides of the detector, consisting of nine disk layers each at z-positions ranging from $|z| = 853.8$ mm to $|z| = 2720.2$ mm. Together with these additional 1976 detector modules in the end-cap regions, the SCT consists of 4088

silicon detector modules in total, covering a pseudo-rapidity region of up to $|\eta| < 2.5$ with a sensitive detection surface of about $63\,\mathrm{m}^2$.

Unlike the pixel detector, the SCT is based on mono-dimensional silicon strips, which, on their own, can only provide one-dimensional track hit information. Therefore, double layers of strips rotated by a stereo angle of $\pm 20\,\mathrm{mrad}$ around the geometrical center of the sensors are employed to allow the reconstruction of two-dimensional track hit measurements on the sensor. Each sensor comprises 768 strips with a length of 12.8 cm per side that are separated by $80\,\mu\mathrm{m}$, resulting in a total number of more than 6 million read-out channels for the entire SCT sub-detector system.



(a)



(b)

Figure 3.6: Schematic illustration of the structural elements of the (a) ATLAS Inner Detector barrel section with its sub-detector systems and the beam pipe [63] as well as (b) a section of one of the two ID end-cap systems including the Pixel and SCT barrel part without IBL [52]. The red lines indicate particles emerging from the interaction region and traversing the detector components.

Transition Radiation Tracker

The Transition Radiation Tracker (TRT) [64] forms the outermost part of the ATLAS Inner Detector, extending from a radius of 563 mm to 1066 mm with a coverage in pseudo-rapidity of $|\eta| < 1.0$ and $1.0 < |\eta| < 2.0$ in its barrel and two end-cap sections, respectively. Unlike the silicon-based Pixel and SCT sub-detector systems, the TRT is a gaseous proportional drift tube detector, consisting of 52 544 gas-filled straws in the barrel part and 122 880 straws in each end-cap region, each of them having a diameter of about 4 mm. While the straws in the barrel section are oriented parallel to the beam axis and 144 cm in length, the straws in the two end-cap regions are 37 cm long and radially arranged with respect to the beam axis. A gold-plated tungsten anode wire with a diameter of 30 μm is located in the center of each drift tube, which is kept at ground potential, while an electric potential of -1.5 kV is applied to the outer tube walls. A charged particle traversing the gas-filled drift tubes creates free electrons through the ionization of gas atoms that are accelerated towards the anode wire. This results in an avalanche of further electrons that amplify the signal which can subsequently be read out by dedicated electronics. Furthermore, the regions between individual straws are filled with special material such as polymer fiber that causes the incident particle to produce additional transition radiation photons, which in turn also ionize the straw's gas and lead to larger signals in the anode wire.

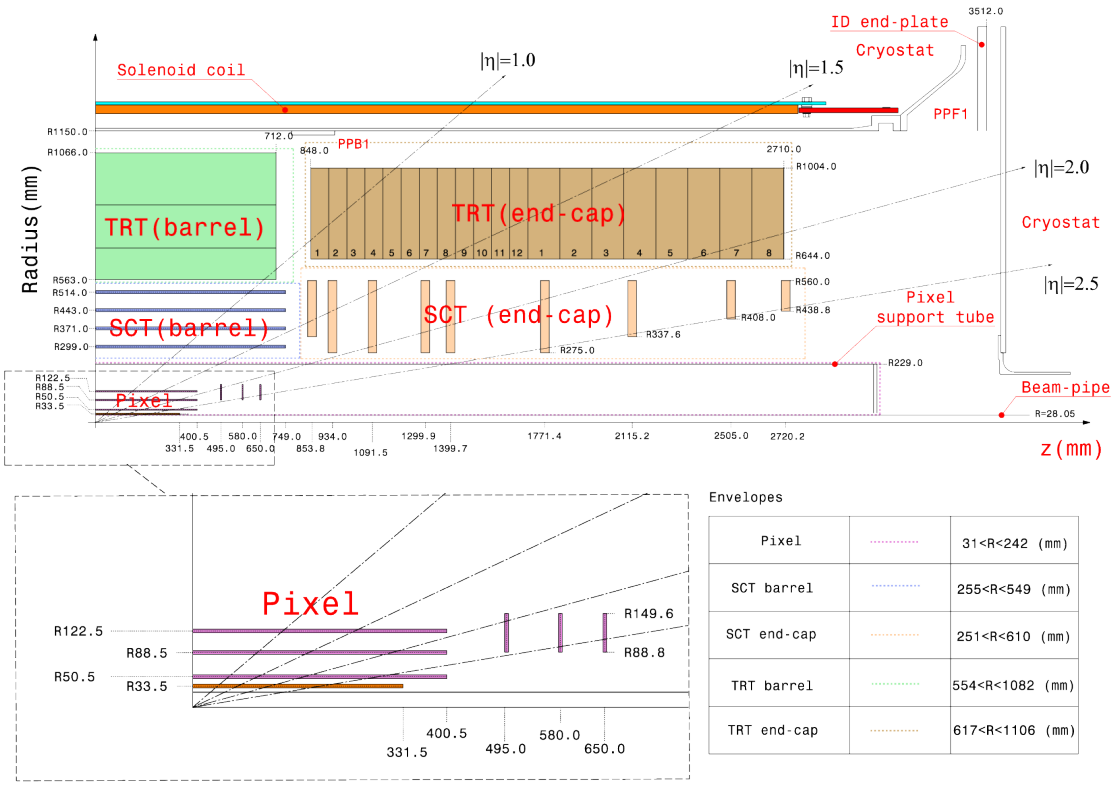


Figure 3.7: Schematic r-z view of a quadrant of the ATLAS Inner Detector. The upper panels shows the layout of the whole ID, whereas a magnified view of the pixel detector as well as the envelopes of all sub-detector systems are given in the bottom panel [65].

Since the energies of transition radiation photons, and therefore also their induced signals, are proportional to the Lorentz factor of the traversing particle, the read-out pulse size can be used to distinguish electrons (which usually come with higher Lorentz factors due to their lower mass) from heavier particles such a pions or muons and therefore complement the electron identification in the calorimeter system.

3.2.3 Calorimeter System

The ATLAS calorimeter system [66, 67], shown in Fig. 3.8, is dedicated to energy measurements of electromagnetically and hadronically interacting particles and plays an important role in e.g. the determination of missing transverse energy. It is situated outside of the Inner Detector and the central solenoid magnet and consists of an electromagnetic as well as a hadronic calorimeter. Each of which is specifically designed to cause the traversing particles to interact with the dense detector material and produce electromagnetic and hadronic particle showers that deposit their energies in the respective calorimeter parts.
The relative energy resolution of a calorimeter system is generally given by

$$\left(\frac{\Delta E}{E}\right)^2 = \left(\frac{c_s}{\sqrt{E}}\right)^2 + \left(\frac{c_n}{E}\right)^2 + c_c^2, \tag{3.8}$$

where the coefficients c_s, c_n and c_c describe stochastic, noise-induced as well as a constant term, respectively. The stochastic term is proportional to $1/\sqrt{E}$ as the number of shower particles increases with larger energies. While the noise-induced term is usually negligible for very high particle energies, the constant term needs to be accounted for, as it includes uncertainties for shower tails that are not fully captured by the detector [68].
Apart from providing precise energy measurements, specific information from the calorimeter system is also used, among others, for online event selection in the ATLAS trigger system (see Section 3.2.5).
A short overview of the design and functionality of the calorimeter sub-detector components will be given in the following.

Electromagnetic Calorimeter

The electromagnetic calorimeter (ECAL) [66] is specifically designed for the measurement of electron- and photon-initiated electromagnetic showers. It is comprised of a barrel and end-cap system, in which alternating layers of dense absorber material, i.e. the material that causes the particle showers, and active material, the material that is used to measure the deposited energies, are deployed to fully capture the desired showers. Calorimeters that utilize alternating layers of absorber and active materials are referred to as *sampling* calorimeters.
The ATLAS ECAL uses lead (Pb) as the absorber material combined with the active material liquid argon (LAr), in which the amount of deposited energy is proportional to the induced LAr ionization signal. These signals can be read out by dedicated electronics, arranged in up to three longitudinal sampling layers with varying granularity from

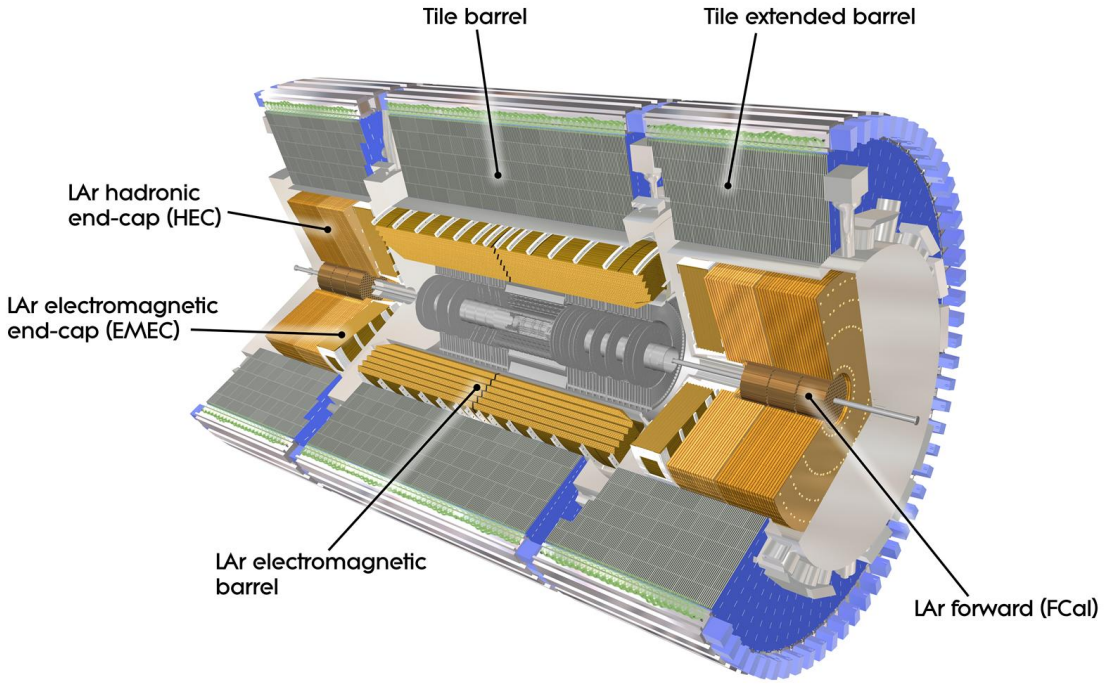


Figure 3.8: Schematic view of the ATLAS calorimeter system [69].

(0.003×0.1) to (0.05×0.025) in $(\Delta\eta \times \Delta\phi)$.
A pseudo-rapidity range of $|\eta| < 1.475$ is covered by the barrel section while the end-caps cover a region of $1.375 < |\eta| < 3.2$.

Hadronic Calorimeter

The hadronic calorimeter (HCAL) system [67] is dedicated to energy measurements of hadrons produced in quark- or gluon-induced particle jets as well as decays of τ leptons or other hadron decays. Similar to the electromagnetic calorimeter, the HCAL is built as a sampling calorimeter and also consists of a barrel and end-cap system. While the barrel tile calorimeter uses steel and special scintillators as the detector material, the hadronic end-cap combines LAr with copper.
Each calorimeter cell has a granularity of $(\Delta\eta \times \Delta\phi) = (0.1 \times 0.1)$. The barrel part covers a pseudo-rapidity range of $|\eta| < 1.7$ while a region of $1.5 < |\eta| < 3.2$ is covered by the end-caps.

Forward Calorimeter

The forward range with $3.1 < |\eta| < 4.9$ is covered by the forward calorimeter (FCAL), which is a combination of an electromagnetic and hadronic calorimeter system. The first of its three layers is optimized for electromagnetic energy measurements, the second and third layer for hadronic interactions. Copper (Cu) and tungsten (W) are used as the absorber, LAr as the active material.

A summary of deployed detector materials, coverage as well as energy resolution of all ATLAS calorimeter subsystems is given in Table 3.1.

Calorimeter		Material	Coverage	Energy resolution	
				$c_s(\%)$	$c_c(\%)$
ECAL	barrel	LAr + Pb	$\lvert\eta\rvert < 1.475$	10	0.7
	end-cap	LAr + Pb	$1.375 < \lvert\eta\rvert < 3.2$	10	0.7
	FCAL	LAr + Cu	$3.1 < \lvert\eta\rvert < 4.9$	28.5	3.5
HCAL	barrel	Scint. + steel	$\lvert\eta\rvert < 1.7$	52	3
	end-cap	LAr + Cu	$1.5 < \lvert\eta\rvert < 3.2$	84	3
	FCAL	LAr + W	$3.1 < \lvert\eta\rvert < 4.9$	94	7.5

Table 3.1: Overview of the main ATLAS calorimeter properties. Energy resolution figures taken from Ref. [68] and Ref. [70].

3.2.4 Muon Spectrometer and Toroid Magnet System

The muon spectrometer (MS) [71] forms the outermost and largest detector system in ATLAS (see Fig. 3.4), extending to a radius of up to $r = 11\,\text{m}$ and is designed to measure the trajectory and momentum of muons in a pseudorapidity region of $|\eta| \leq 2.7$.
It consists of four gas-based sub-detector systems, two of which are located in the central barrel and two in the forward region. In each region, one of the two systems is dedicated to precision momentum measurements while the other features fast response times with coarser granularity and is used for the ATLAS trigger system (see Section 3.2.5).
The precision measurements are performed by so-called monitored drift tubes in the barrel region, covering a pseudo-rapidity range of $|\eta| < 2.0$, and are complemented by cathode strip chambers in the forward region with a coverage of $2.0 < |\eta| < 2.7$.
The muon trigger chambers consist of resistive-plate chambers in the barrel, covering a region of $|\eta| < 1.05$, and thin-gap chambers in the forward region with a coverage of $1.05 < |\eta| < 2.4$.
A superconducting toroid magnet system is used to provide strong magnetic fields for bending the muon's trajectories and hence allowing precise momentum measurements. It is situated outside the calorimeter and within the ATLAS muon system and provides magnetic fields of up to 4 T. It consists of eight barrel coils, each of which weighing about 100 t, and two end-cap systems. The toroid magnet system is operated at a nominal temperature of 4.7 K.
Although the muon spectrometers could be used standalone for the reconstruction and identification of muons, additional tracking information from the ID is usually incorporated as it significantly improves the sub-detector's overall performance.

3.2.5 Trigger and Data Acquisition

With the LHC delivering collisions at a design bunch-crossing rate of about 40 MHz and the ATLAS detector only being able to write out events to permanent storage at a rate of approximately 1 kHz, a two-level trigger system [72] is used for online event selection

to reduce the read-out data stream to manageable levels. Since most of the interesting physics processes are rare with cross-sections orders of magnitudes lower than the one of inelastic proton-proton scattering, the ATLAS trigger system needs to efficiently select those collision events that will be of potential interest for further offline analyses.

The trigger system used in ATLAS during LHC Run 2 is shown in Fig. 3.9. It consists of a hardware-based first level trigger (Level-1), which reduces the event rate from the LHC bunch crossing rate to approximately 100 kHz, as well as a second stage software-based high level trigger (HLT), further reducing the Level-1 output rate to the ATLAS read-out rate of about 1 kHz.

In addition to the initial event rate reduction, the Level-1 trigger also determines so-called Regions of Interest (RoIs) in the detector based on coarse granularity calorimeter and muon detector information. The HLT can subsequently run sophisticated selection algorithms on full granularity detector information in these RoIs or even on the entire event. While the decision time for a Level-1 accept is 2.5 µs, the HLT processing time is about 200 ms. Once an event has passed both trigger stages, and therefore most likely contains interesting physics processes, it will be written to data storage for further analysis.

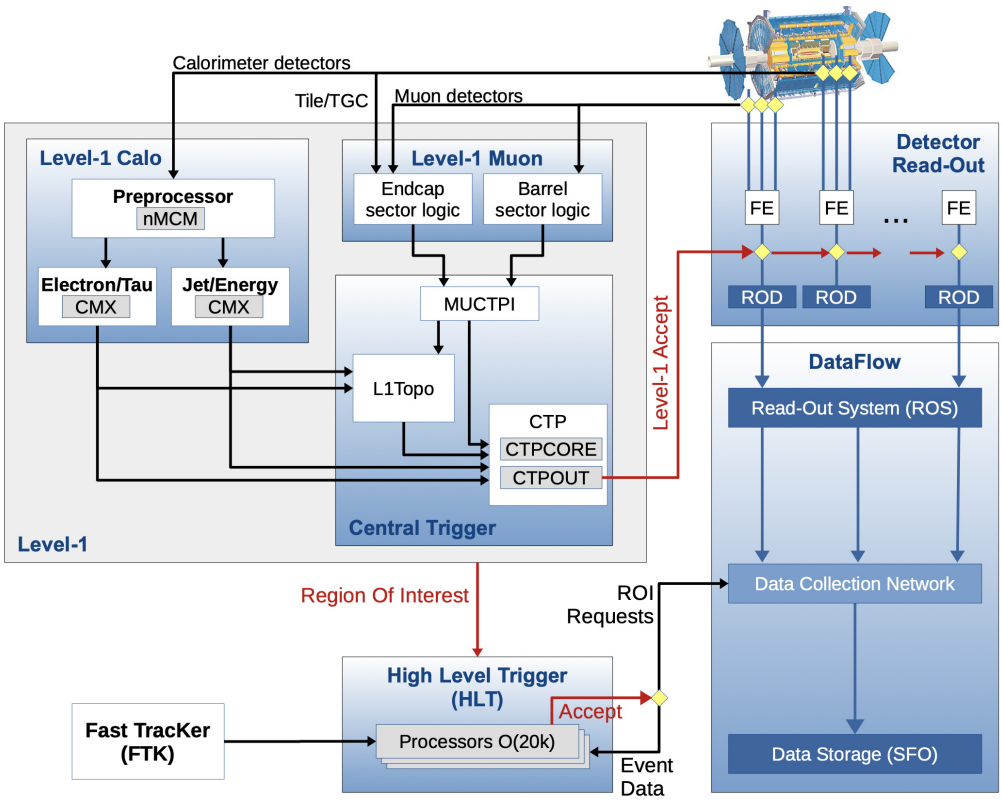


Figure 3.9: Schematic layout of the ATLAS trigger and data acquisition system as used in Run 2 [72].

3.3 ATLAS Phase-2 Upgrade for HL-LHC

In order to prepare for the challenging HL-LHC conditions with expected pile-up values of up to $\langle\mu\rangle = 200$ (see Section 3.1.3), several ATLAS sub-detector systems as well as all major software components will need to undergo significant upgrades. In the following, two of the planned hardware upgrades, the new Inner Tracker and the new High-Granularity Timing Detector, as well as the resulting software challenges will be briefly discussed.

3.3.1 Inner Tracker (ITk)

During the ATLAS Phase-2 upgrade for HL-LHC, the entire ATLAS Inner Detector will be replaced by the new, all-silicon Inner Tracker (ITk) [73, 74]. The ITk is designed to improve the resolution and radiation hardness of the ATLAS tracker under the severe upcoming pile-up conditions, by exploiting new sensor and read-out electronics technologies as well as a novel dedicated detector design, as schematically illustrated in Fig. 3.10.
Its layout consists of a silicon pixel detector, the innermost part of the ITk, instrumented with high-granularity silicon pixels of a proposed size of either 50 µm × 50 µm or 25 µm × 100 µm. The pixel detector is enclosed by a silicon strip detector system with strip lengths of 24.1 mm in the innermost layers and 48.2 mm in the outermost layers where the expected occupancy will be lower.

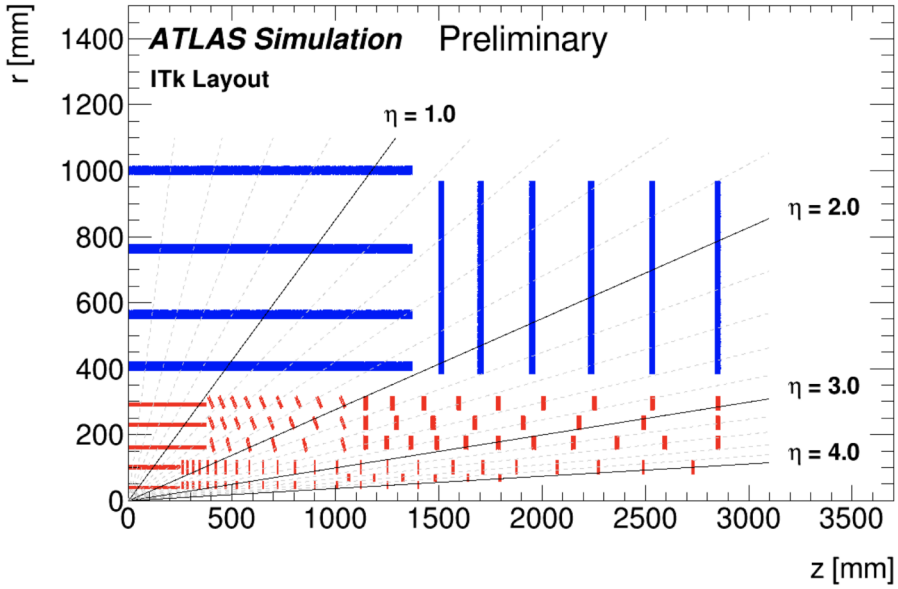


Figure 3.10: Schematic depiction of the ITk detector layout [75]. The active elements of the pixel sub-detector system are shown in red, those of the strip sub-detector system are shown in blue.

The pixel subsystem comprises five flat barrel layers as well as five layers of inclined or vertical rings which together provide a coverage of up to $|\eta| < 4$. The strip sub-detector system consists of four strip module layers in the barrel region as well as six disks in the end-caps, resulting in a $|\eta| < 2.7$ pseudo-rapidity coverage.

The considerable high-granularity extension of tracking coverage in the forward region compared to the ATLAS ID will, in addition to providing a significantly better efficiency and acceptance in the forward region, also complement a large part of the forward calorimeter systems and therefore enables an enhanced reconstruction of forward physics objects.

3.3.2 High-Granularity Timing Detector (HGTD)

The high luminosity environments of HL-LHC will lead to pile-up densities of up to 4 primary vertices/mm along the beam axis. Correctly identifying the hard-scatter vertex as well as associating the correct tracks, and thus physics objects, to it will be a major challenge in these conditions.

The High-Granularity Timing Detector (HGTD) [76] is a novel detector system which will be introduced to ATLAS during the Phase-2 upgrade for HL-LHC. It augments the ITk detector with the capability to measure charged particle tracks in the forward region not only in space but also in time with a timing resolution of less than 50 ps. Primary interaction vertices that are overlaid in space but separated in time, as shown in Fig. 3.11, can therefore be better resolved and reconstructed separately, allowing to mitigate the influence of pile-up vertices.

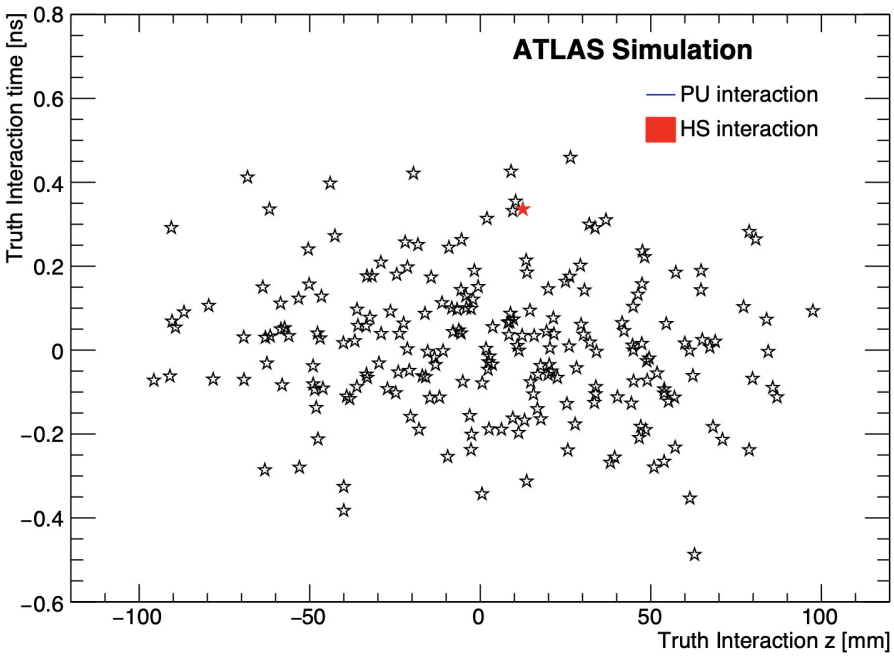


Figure 3.11: Visualization of the truth primary interaction vertices during a single $\langle\mu\rangle = 200$ bunch crossing in the z–t plane, showing the simulated hard-scatter vertex as well as several pile-up interactions [76]. Note that the targeted HGTD resolution of about 50 ps roughly corresponds to the marker size used in the plot. While a timing measurement will clearly help to better reconstruct some of the vertices, the HGTD will not be able to efficiently resolve vertices that are too close to one another in the depicted z–t plane.

With a radius between 120 mm and 640 mm at a position of $z = 3.5$ m along the beam axis, the two HGTD disks will be placed between the ITk and the calorimeter end-caps, as illustrated in Fig. 3.12, covering a pseudo-rapidity range of $2.4 \leq |\eta| \leq 4$.
In addition to improving the reconstruction efficiency of physics objects in the forward region, the High-Granularity Timing Detector can also be used for online and offline luminosity measurements and therefore augments the existing ATLAS luminosity measurement sub-detector systems (see e.g. Ref. [77]).

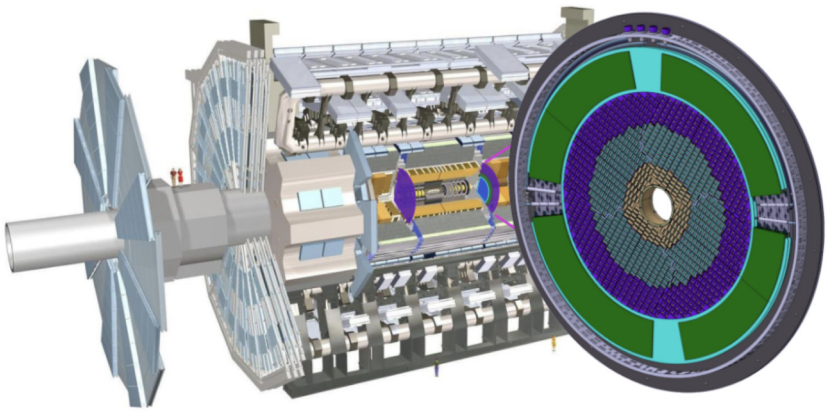


Figure 3.12: Schematic illustration of the HGTD position within the ATLAS detector [76].

The current ATLAS Run 2 software for the reconstruction of primary vertices does not provide any functionality or mathematical concepts to include timing information in the vertex reconstruction process. As this functionality will be crucial in the future, new concepts for time-dependent vertex reconstruction as well as an entirely new vertex reconstruction software suite, which allows to natively include timing information, have been developed in the context of this thesis and will be presented in Chapter 5 and 6, respectively.

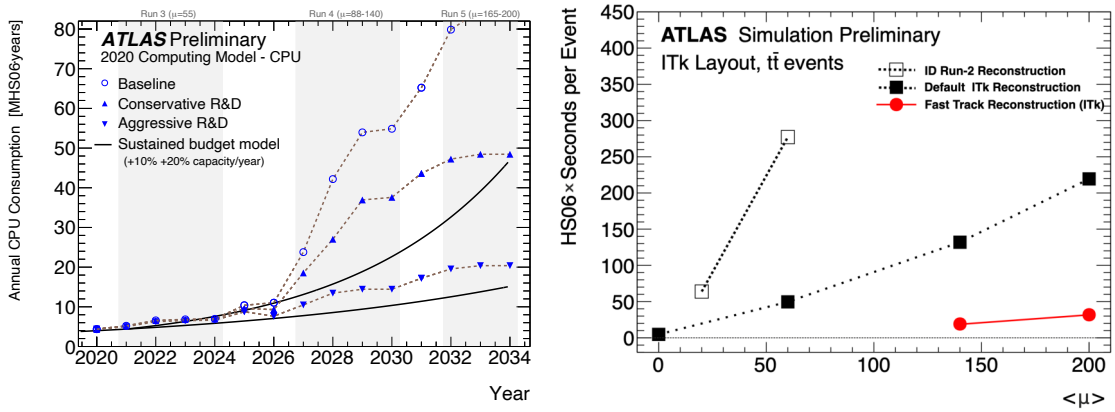
3.3.3 Software Challenges in High-Luminosity Environments

The high-luminosity environments expected in the HL-LHC era will result in collision events with unprecedented complexity while additionally the expected ATLAS data readout rate will increase by approximately a factor of ten to about 10 kHz. In order to cope with the resulting challenges in terms of computational capacity and available disk space, while respecting the associated computing budget constraints, very significant software and computing improvements will be inevitable before the start of LHC Run 4, planned to take place in the year 2027 [78].
Fig. 3.13(a) shows the estimated required ATLAS CPU resources for the upcoming years under different R&D scenarios as well as the estimated available resources assuming a sustained budget model. While the required CPU resources for a Run 3 baseline model projected to Run 4 and Run 5 conditions would immensely exceed the expected available resources, both a conservative and especially an aggressive R&D scenario will significantly

decrease the required resources to more manageable levels. Therefore, enormous R&D efforts have to be undertaken for all major software and computing components, such as the ATLAS track reconstruction software, one of the main contributors to the overall needed CPU resources.

The expected track reconstruction times per event in different pile-up environments for the ATLAS ID Run 2 reconstruction as well as for the default ITk and a fast track ITk reconstruction is shown in Fig. 3.13(b). While the ID Run 2 track reconstruction scales drastically with an increase in pile-up, the default ITk reconstruction exhibits – although utilizing a very similar reconstruction strategy and software as the ID Run 2 reconstruction – a more favorable scaling, mainly due to the improved and dedicated all-silicon ITk detector architecture. Ongoing studies on fast track ITk reconstruction can achieve significant performance speed-ups in high pile-up environments when modifying the reconstruction workflow, result however also in a degraded physics performance compared to its non-optimized counterpart [79].

While many performance optimization campaigns and projects for various reconstruction components in HL-LHC environments already exist, such as the above mentioned fast track ITk reconstruction, no such optimization efforts have yet been undertaken for primary vertex reconstruction algorithms, whose CPU performances are also expected to degrade immensely in the upcoming challenging conditions. A major part of this thesis will therefore deal with the development and implementation of a new, modern and highly optimized vertex reconstruction software suite (see Chapter 6) as well as the development of novel techniques for vertex seed finding in HL-LHC environments (see Chapter 5), one of the computationally most expensive components in vertex reconstruction in these high pile-up conditions.



(a) (b)

Figure 3.13: (a) Estimated required CPU resources needed for the years 2020 to 2034 under a (Run 3) baseline, conservative R&D and aggressive R&D scenario. The solid lines indicate a 10% and 20% annual improvement in hardware capacity, assuming a sustained budget level [80]. (b) Expected track reconstruction times in different $\langle\mu\rangle$ environments for Run 2, default and fast ITk track reconstruction software [79].

3.4 Simulation and Data Processing in the Athena Software Framework

The ATLAS event processing software framework Athena [81] is used for almost all ATLAS computing and production workflows. It is designed to both reconstruct data collected by the ATLAS detector as well as to produce simulated Monte Carlo (MC) samples, which are of vital importance for all physics analyses as well as for testing and validation of event processing algorithms and tools.
A typical ATLAS workflow for the production of simulated MC samples in Athena can be summarized as follows [81, 82]:

1. **Event generation:** The very first step in the Monte Carlo simulation and reconstruction chain is the generation of the physics event, in which a hard-scatter collision as well as overlaid minimum bias proton-proton collisions are simulated by an event generator such as Pythia [83] or Sherpa [84]. The event generation typically involves the matrix element calculation, describing the transition amplitude between the initial and final state up to a certain order in perturbation theory, the simulation of higher order effects using parton shower algorithms up to a so-called resummation scale μ_Q, an unphysical scale introduced to define an upper cutoff on the parton shower evolution, as well as the subsequent hadronization processes [85, 86]. In order to avoid infrared and ultraviolet divergences during matrix element calculation, unphysical factorization and renormalization scales μ_F and μ_R are introduced, for which, similar to the resummation scale μ_Q, no values can be derived from first principles. In fact, the exact choice of these scales highly affects the simulation outcome, typically making it a major source of systematic uncertainty in physics analyses. Double counting of hard partons described by matrix element calculation and more soft collinear partons covered by the parton shower algorithm is avoided using dedicated methods, such as the CKKW technique [87], which introduce an unphysical *matching scale* and therefore another source of systematic uncertainty to the event generation process.
   A list of all generated particles together with their momentum four-vectors is produced and subsequently passed to the next step.

2. **Detector simulation:** In the second stage, the interactions of the generated particles with the detector material need to be simulated. The Geant4 simulation toolkit [88] is used to perform this task, exploiting a detailed ATLAS detector geometry description and the knowledge of the material distribution within the detector. The particles are transported through the applied magnetic field and various material interactions such as multiple scattering, energy loss or photon conversions as well as decays of unstable particles are simulated.

3. **Detector response:** The detector response simulation stage, also called *digitization*, emulates the detector response, and thus its electronics output, for every single particle hitting a sensitive detector element. The output of the digitization step is

of the same format as real data collected by the experiment and can therefore be similarly processed in the following reconstruction step.

4. **Reconstruction:** Based on the detector response from either real data or simulation, numerous algorithms are deployed to reconstruct the physics event. This includes the reconstruction of charged particles trajectories (so-called *tracking*) and primary as well as secondary interaction vertices (so-called *vertexing*), the clustering of particle jets in the calorimeters, the reconstruction and identification of muons as well as many other downstream algorithms that help to identify properties of the collision event.
   As a large part of the given thesis deals with the development and implementation of primary vertex reconstruction software, detailed descriptions of tracking and vertexing algorithms are presented in Chapter 4 and Chapter 5, respectively. Details on jet clustering algorithms as well as the reconstruction and identification of muons will be given in Section 9.4.

Based on all reconstructed quantities in a collision event, physics objects such as electrons, photons, muons or (heavy-flavor) jets can be defined for a physics analysis. Details on the exact definitions of physics objects will be discussed in the context of the presented analysis in Chapter 9.

Part II

Algorithms and Software for Primary Vertex Reconstruction

Chapter 4

Reconstruction of Charged Particle Trajectories

4.1 Introduction

The reconstruction of charged particle trajectories, also referred to as *tracking*, is a key component in the event reconstruction of almost any HEP experiment. Based on a set of measurements (so-called *hits*) resulting from the interactions of charged particles with dedicated sensitive detector systems, such as the ATLAS Inner Detector or Muon Spectrometer tracking systems (see Section 3.2.2 and Section 3.2.4, respectively), the goal of tracking is to find groups of hits within the input measurement set that are expected to have originated from the same particles and estimate the parameters describing their associated trajectories, the so-called *track parameters.*
These reconstructed charged particle trajectories, or *tracks*, are vital for many downstream reconstruction algorithms, the most immediate one being the reconstruction of primary interaction vertices (see Chapter 5). As a large part of this PhD thesis deals with software and algorithmic developments for the reconstruction of primary interaction vertices, a more detailed description of track reconstruction, especially in the context of ATLAS, will be given in this chapter.

4.2 ATLAS Track Reconstruction

The strategy for track reconstruction in the ATLAS Inner Detector [63] consists of several subsequent stages and can be summarized as follows. In a first step, clusters of pixels and strips with deposited charge above a certain threshold are formed in the ID Pixel and SCT detectors, from which three-dimensional space points, representing the position where the particle traversed the detector, can be created. Track seeds can subsequently be built from sets of three space points, which provide enough directional information to follow the potential trajectories and augment the track seeds with more compatible measurements in order to form track candidates. This track candidate finding and fitting procedure is performed using a combinatorial Kalman filter. Ambiguities in the found track candidate collection are then resolved based on a score that is assigned to each track candidate,

resulting in a final collection of reconstructed tracks that can subsequently be extended to incorporate measurements from the TRT in order to improve their momentum resolution. The above outlined individual steps will be separately discussed in more detail in the following.

4.2.1 Clusterization

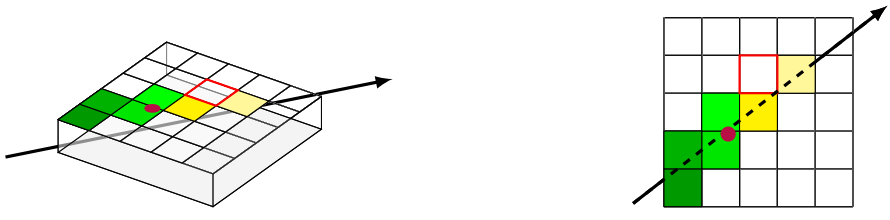
Starting from raw detector measurements, an estimated position where the particle traversed the detector element needs to be extracted. For this purpose, clusters of signals from adjacent channels with deposited charge above a certain threshold in the Pixel and SCT subsystems are formed by iteratively grouping neighboring segments using a Connected Component Analysis [89] approach.
The effective cluster position can subsequently be determined. The most straight-forward approach is to calculate the average geometric segment position as

$$\vec{x} = \sum_{i=1}^{N} \vec{l_i}, \tag{4.1}$$

where N is the number of segments in the cluster of interest and $\vec{l_i}$ denotes the i-th local cluster position. Alternatively, the charge-weighted average cluster position can be determined as

$$\vec{x} = \frac{1}{\sum_{i=1}^{N} q_i} \sum_{i=1}^{N} q_i \vec{l_i}, \tag{4.2}$$

where the collected charge in segment i is given by q_i.
Novel techniques for the determination of the cluster positions also rely on the deployment of artificial neural networks that can additionally estimate the corresponding position uncertainties as well as the probability that the cluster was created by several collimated particles, allowing to potentially split the merged cluster [90].
A schematic illustration of a charged particle traversing a pixel sensor as well as the resulting pixel cluster with the estimated average pixel position is shown in Fig. 4.1.



(a) (b)

Figure 4.1: Schematic illustration of the clustering of multiple pixels that collected charge above a certain threshold, caused by a charged particle traversing the detector element. A three-dimensional view is presented in (a), whereas (b) shows a projection onto a two-dimensional plane. Different colors indicate different amounts of collected charge. The red framed pixel did not receive charge above threshold and is therefore not considered for clusterization. A charge-weighted average cluster position is illustrated by a purple dot.

Several pixels detected charge above the read-out threshold (depicted as colored pixels) and are eligible for pixel clustering. One single pixel (shown with a red frame) did also detect charge, the total amount collected, however, was too low for the pixel to be considered for clusterization.

4.2.2 Space Point Formation and Seeding

So far, the measurement positions are only given in the local detector surface reference frame and need to be converted into global, three-dimensional space points. For a pixel sensor measurement, detector geometry information such as the sensor's position and rotation are used to determine the global measurement position from the local, two-dimensional cluster position. Since the ATLAS SCT strip modules always come in pairs, superimposed with a stereo angle between them, their one-dimensional measurements can be combined to determine a two-dimensional local measurement position, which can subsequently be used to form a three-dimensional space point.
The following track finding procedure starts with the formation of so-called *track seeds* which consist of space point triplets originating from unique detector layers with increasing radii, aiming to provide initial estimates of groups of measurements that belong to a common track [91], as schematically depicted in Fig. 4.2. The default number of three required space points allows for a high number of possible space point combinations, while additionally a rough estimate of the track seed's momentum can be obtained. Also, under the assumption of a perfectly helical track model in a constant magnetic field, the seed's transverse impact parameters, i.e. the minimal transverse distance of the estimated trajectory to the beam axis (see Section 4.3), can be estimated in order to ensure that the seed is compatible with a track that originated from the pp interaction region in the center of the detector.

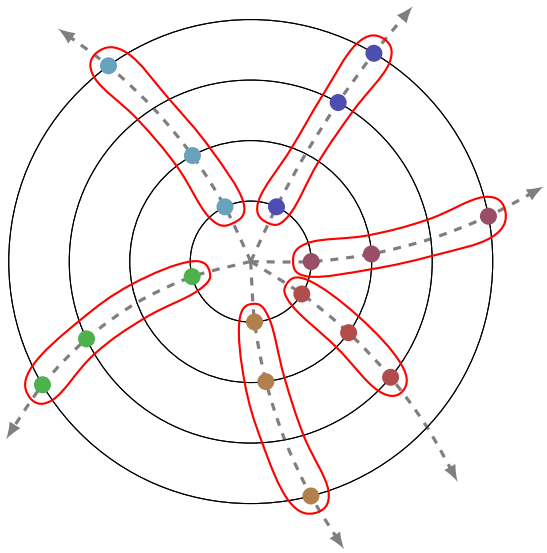


Figure 4.2: Schematic illustration of a number of space points on different detector layers in the transverse plane, used to form various initial track seeds of potential charged particle trajectories being compatible with having originated from the detector center.

4.2.3 Track Finding and Track Fitting

Track seeds that pass the initial seed quality requirements, such as a minimum estimated transverse momentum and an interaction region compatibility criterion, are used as input for the following track finding and fitting procedure, which is based on a combinatorial Kalman filter technique that is deployed by ATLAS in its track seeding and following approach. The Kalman filter formalism [92, 93], originally developed for monitoring real-time mechanical processes, is a commonly used method for track fitting in high energy physics and will be briefly discussed in the following.

The Kalman Filter

The Kalman filter describes a linear dynamic system, where its state, representing the parametrization of the particle trajectory, at every step k is given by the state vector $\vec{x}$. In the context of track fitting, each step k is a new measurement to be incorporated into the fitting procedure.
The evolution of the state vector from step $k-1$ to step k is then given by

$$\vec{x}_k = \mathbf{F}_{k-1}\vec{x}_{k-1} + \vec{w}_{k-1}, \tag{4.3}$$

where the linear propagation function $\mathbf{F}_{k-1}$ describes the transport from the previous state vector $\vec{x}_{k-1}$ at step $k-1$ to the current one at step k and $\vec{w}_{k-1}$ denotes a random process noise contribution term that is added to the system during the propagation step.
At every step k a new measurement $\vec{m}_k$ is added to the system which can be described by a linear function of the state vector at step k with an associated Gaussian noise term $\vec{\epsilon}_k$:

$$\vec{m}_k = \mathbf{H}_k\vec{x}_k + \vec{\epsilon}_k \tag{4.4}$$

Here, $\mathbf{H}_k$ denotes the so-called *measurement mapping function* that maps a particle's trajectory parameters to an actual measurement on a detector surface.
Given a dynamic linear system as described above, the Kalman filter track fit follows a three-step procedure:

1. **Prediction:** First, based on all available information at the current step, and therefore incorporating all previous measurements, the state vector at the next step is predicted. The predicted state vector at any step k can therefore be written as a function of the previous step $k-1$ as

   $$\vec{x}_k^{k-1} = \mathbf{F}_{k-1}\vec{x}_{k-1}. \tag{4.5}$$

   Additionally, the covariance matrix $\mathbf{C}_{k-1}$, describing the track state's uncertainties at step $k-1$, must be transported and predicted at step k:

   $$\mathbf{C}_k^{k-1} = \mathbf{F}_{k-1}\mathbf{C}_{k-1}\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1} \tag{4.6}$$

Here, the noise term $\mathbf{Q}_{k-1}$ describes stochastic effects like multiple Coulomb scattering of the particle traversing detector material and therefore corresponds to the covariance matrix of $\vec{w}_{k-1}$ in Eq. (4.3).

2. **Filtering:** Taking the actual measurement $\vec{m}_k$ as well as the predicted state vector $\vec{x}_k^{k-1}$ at step k into account, the new updated (*filtered*) state vector $\vec{x}_k$ at step k can be determined by

$$\vec{x}_k = \vec{x}_k^{k-1} + \mathbf{K}_k \left(\vec{m}_k - \mathbf{H}_k \vec{x}_k^{k-1} \right), \tag{4.7}$$

where the adaptation of the predicted state vector $\vec{x}_k^{k-1}$ with respect to the measurement $\vec{m}_k$ is governed by the so-called *Kalman gain matrix*[1] $\mathbf{K}_k$:

$$\mathbf{K}_k = \mathbf{C}_k^{k-1} \mathbf{H}_k^{\mathrm{T}} \left[\mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k^{k-1} \mathbf{H}_k^{\mathrm{T}} \right]^{-1} \tag{4.8}$$

Here, $\mathbf{V}_k$ denotes the covariance matrix associated with the measurement error term $\vec{\epsilon}_k$ in Eq. (4.4) at step k and therefore effectively represents the $\vec{m}_k$ measurement's uncertainty.
In addition to the filtering of the state vector as described above, the associated predicted covariance matrix can also be updated. The filtered covariance matrix at step k is then given by

$$\mathbf{C}_k = (\mathbb{1} - \mathbf{K}_k \mathbf{H}_k) \mathbf{C}_k^{k-1}. \tag{4.9}$$

3. **Smoothing:** The last stage of the Kalman formalism is the smoothing stage, in which the state vector at step k can be further improved, taking information from the subsequent step $k+1$ into account. In practice, the smoothing is only performed after all available measurements have been added to the system by the above described prediction and filtering stages. This allows to update all state vectors successively in reverse, including measurement information from all subsequent steps. Considering a filtered track candidate with n associated measurements, the smoothed state vector at step k is given by

$$\vec{x}_k^n = \vec{x} + \mathbf{A}_k \left(\vec{x}_{k+1}^n - \vec{x}_{k+1}^k \right), \tag{4.10}$$

where $\vec{x}_{k+1}^n$ denotes the previously smoothed state vector at step $k+1$, $\vec{x}_{k+1}^k$ the predicted state vector at step $k+1$ and $\mathbf{A}_k$ the smoothing gain matrix

$$\mathbf{A}_k = \mathbf{C}_k \mathbf{F}_k^{\mathrm{T}} \left(\mathbf{C}_{k+1}^k \right)^{-1} \tag{4.11}$$

with the predicted covariance matrix $\mathbf{C}_{k+1}^k$ at step $k+1$.
Lastly, also the covariance matrix at step k is smoothed and can be written as

$$\mathbf{C}_k^n = \mathbf{C}_k + \mathbf{A}_k \left(\mathbf{C}_k^n - \mathbf{C}_{k+1}^k \right) \mathbf{A}_k^{\mathrm{T}}. \tag{4.12}$$

[1] Another mathematically equivalent technique for filtering exists that will be not be discussed here in the context of track fitting, as ATLAS makes use of the presented Kalman gain formalism. Both techniques are presented in Ref. [93].

A schematic representation of the Kalman filter prediction and filtering stages in the context of track reconstruction is shown in Fig. 4.3. The filtered state at step $k-1$, obtained by taking the measurement $\vec{m}_{k-1}$ as well as the predicted state $\vec{x}_{k-1}^{k-2}$ according to Eq. (4.7) into account, is used to predict the state vector at step k, representing the state at the next detector layer with measurement $\vec{m}_k$. The predicted state vector and measurement at step k are then again used to determine a filtered state. The procedure is repeated until all compatible measurements are included in the fit, followed by the backwards smoothing stage which is not shown for the sake of conciseness.

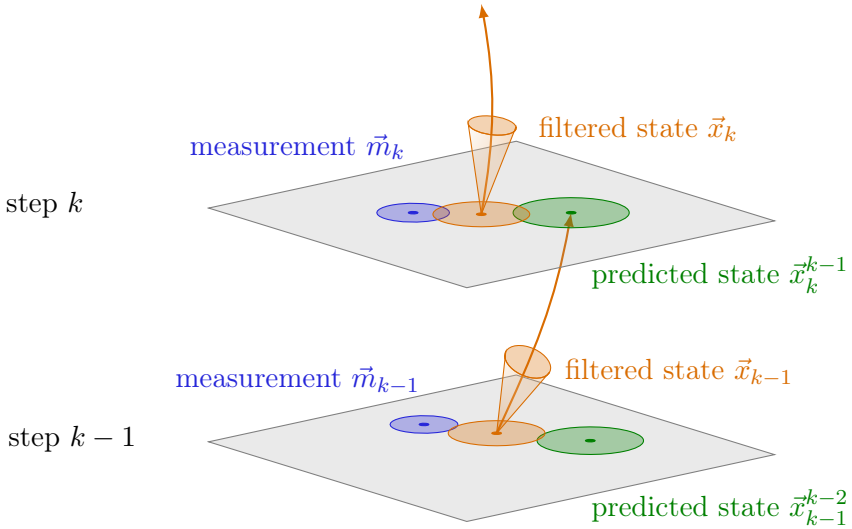


Figure 4.3: Schematic illustration of the Kalman filter prediction and filtering stages, showing the evolution from step $k-1$ to step k and thus representing the incorporation of the measurements $\vec{m}_{k-1}$ and $\vec{m}_k$ on two different detector layers into the track fit. The smoothing step is not shown for the sake of conciseness.

Kalman Filter Application in ATLAS

In practice, the assumption of a linear transport function $\mathbf{F}$ in the Kalman filter formalism does not hold for the propagation of a charged particle in the inhomogeneous magnetic field of the ATLAS Inner Detector. Therefore, the prediction in Eq. (4.5) is retrieved by calculating the intersection of the extrapolated particle trajectory with the next sensitive detector layer. A so-called *extrapolator* (or also *propagator*) engine is used for this task, enabling the precise transport of track parameters and their associated covariance matrices through the detector materials and the applied magnetic field. The ATLAS extrapolator makes use of the Runge-Kutta-Nyström method [94–97] at fourth order to numerically solve the equation of motions of the charged particles traversing the magnetic field.
In addition to the predicted track parameters at a given step, knowledge about the associated covariance matrix at the new position is often essential for many applications. Exploiting the Jacobian $\mathbf{J}_{i\to f}$ for the transformation from some initial track parameter

state $\vec{x}_i$ to some final track parameter state $\vec{x}_f$ with

$$\mathbf{J}_{i\to f} = \frac{\partial \vec{x}_f}{\partial \vec{x}_i}, \tag{4.13}$$

the corresponding *covariance transport* from the initial covariance matrix $\mathbf{C}_i$ to the final covariance matrix $\mathbf{C}_f$ is given by

$$\mathbf{C}_f = \mathbf{J}_{i\to f}\, \mathbf{C}_i\, \mathbf{J}_{i\to f}^{\mathrm{T}}. \tag{4.14}$$

As discussed in Ref. [98], the required Jacobian is obtained in three steps. First, the transformation from the local covariance matrix reference frame to a global frame is found, followed by a semi-numerical method which accumulates effective derivatives between consecutive Runge-Kutta-Nyström integration steps to estimate the total effective Jacobian from the initial to the final global position. Lastly, the back transformation to the local final coordinate system is performed in order to obtain the full Jacobian $\mathbf{J}_{i\to f}$ for the required covariance matrix transport in the ATLAS extrapolator engine.

Also, the actual track finding and fitting method used by ATLAS is an extension of the above described Kalman filter formalism, the so-called *combinatorial Kalman filter* (CKF) [99]. Starting from an initial seed, the CKF does not only consider single measurements for the track fit but allows a branching of the fit at each encountered layer while selecting measurements based on their compatibility with the latest state vector estimate. Multiple track candidates can therefore arise from one single track seed, giving rise to a dedicated ambiguity resolution, which will be explained in the following.

4.2.4 Ambiguity Resolution

Due to its combinatorial nature, the above discussed track finding procedure results in a non-negligible amount of *fake* track candidates, i.e. candidates that do not represent a real particle trajectory but were formed from a random combination of compatible hits in the detector. A dedicated ambiguity resolution step [63] is therefore performed in the final stage of track reconstruction. Here, a track score, largely based on simple track quality measures, is determined for each track candidate and the final reconstructed track collection is assembled with tracks in descending score order, favoring tracks of high quality.
Large numbers of associated hits as well as high track momenta increase the final track score, whereas holes, i.e. absence of an expected measurement of the reconstructed trajectory on a sensitive detector element, and high track χ^2 values lead to a lower score.
Track candidates that meet the following basic quality selection criteria [63]

- $p_t > 400\,\mathrm{MeV}$,
- $|\eta| < 2.5$,
- at least seven Pixel and SCT hits,
- at most one shared Pixel cluster or two shared SCT clusters on the same layer,

- a maximum of two holes in the combined Pixel and SCT system,
- a maximum of one hole in the Pixel detector,
- $|d_0| < 2\,\text{mm}$,
- $|z_0 \sin\theta| < 3\,\text{mm}$,

where d_0 and z_0 denote the transverse and longitudinal impact parameters, respectively (see Section 4.3), are further extended to the TRT in order to improve the track's momentum resolution, resulting in the final output collection of reconstructed charged particle trajectories.

4.3 Track Parametrization

The trajectory of a charged particle in a magnetic field can be fully described in either a purely global reference frame or, more conveniently in the context of many applications such as the reconstruction of interaction vertices, with respect to a reference surface or line with local position coordinates.
In a global representation, the particle's trajectory can be parametrized by a global position $\vec{r} = (x, y, z)^{\mathrm{T}}$, a global momentum $\vec{p} = (p, \phi, \theta)^{\mathrm{T}}$ and its charge q.
The momentum vector can also be expressed as the individual components of the spatial direction unit vector in spherical coordinates

$$\begin{aligned} T_x &= \sin\theta\cos\phi \\ T_y &= \sin\theta\sin\phi \\ T_z &= \cos\theta, \end{aligned} \tag{4.15}$$

where the total momentum value p is then incorporated in a measure of the inverse curvature of the trajectory, q/p. A possible global parametrization, which is also referred to as *free parametrization*, is then given by

$$\vec{x} = (x, y, z, T_x, T_y, T_z, q/p)^{\mathrm{T}}. \tag{4.16}$$

The track parameter vector is also often denoted as $\vec{q} = \vec{x}$. Note that many different conventions exist and the representations can change accordingly.
Furthermore, a temporal component t can also be included if a time-dependent application is desired.
The representation with respect to a local reference frame mainly depends on the type of reference system. In case of a simple planar surface, as often the case in track finding and fitting to describe planar silicon sensors, two local coordinates (l_x, l_y) together with the particle's global momentum and charge are used to provide a track parametrization [100] according to

$$\vec{x} = (l_x, l_y, \phi, \theta, q/p)^{\mathrm{T}}. \tag{4.17}$$

A very often used local track parametrization, also in the context of the reconstruction of interaction vertices, is the so-called *perigee parametrization*. Here, the track's parameters are expressed in the point of closest transverse approach, the perigee, with respect to a reference line, often chosen to be the beam axis. As illustrated in Fig. 4.4, the transverse distance between the perigee point and the reference line is denoted as d_0, while the corresponding z-coordinate is called z_0. These two parameters, d_0 and z_0, are called the transverse and longitudinal impact parameters, respectively, and are used as the two local parameters in the perigee track parametrization:

$$\vec{x} = (d_0, z_0, \phi, \theta, q/p)^{\mathrm{T}} \tag{4.18}$$

Again, an additional temporal component can be included, if desired for a specific time-dependent application.

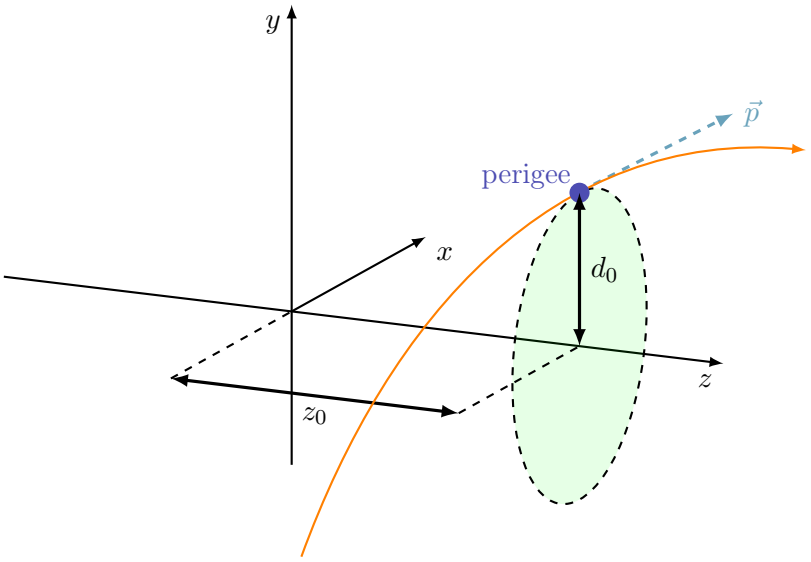


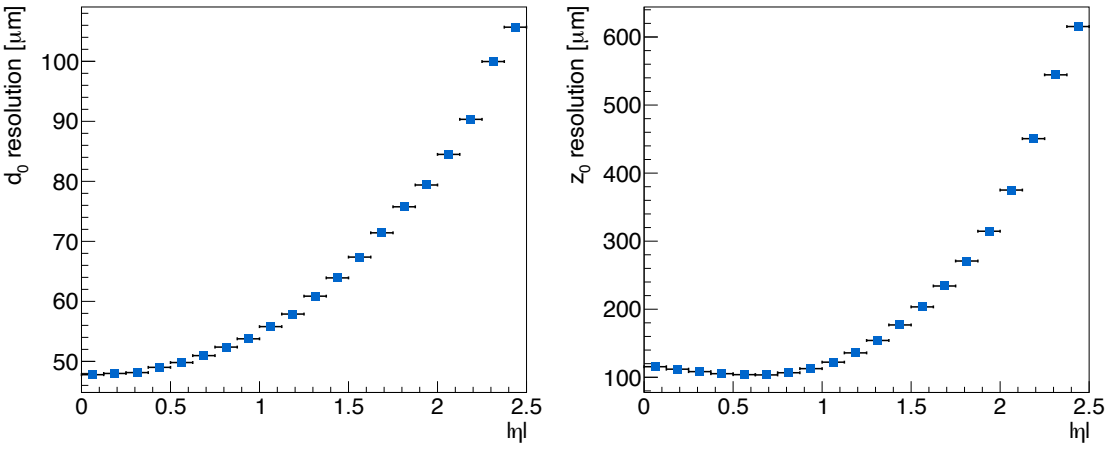
Figure 4.4: Schematic depiction of the (perigee) point of closest transverse approach to the z-axis for a trajectory (orange) of a particle with momentum $\vec{p}$. The resulting impact parameters, d_0 and z_0, are also illustrated.

The associated track parameter uncertainties are expressed in terms of a covariance matrix, which, in case of the perigee parametrization, takes the form

$$\mathbf{C} = \begin{bmatrix} \sigma^2(d_0) & \sigma(d_0, z_0) & \dots & \sigma(d_0, q/p) \\ \sigma(z_0, d_0) & \sigma^2(z_0) & \dots & \sigma(z_0, q/p) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(q/p, d_0) & \sigma(q/p, z_0) & \dots & \sigma^2(q/p) \end{bmatrix}. \tag{4.19}$$

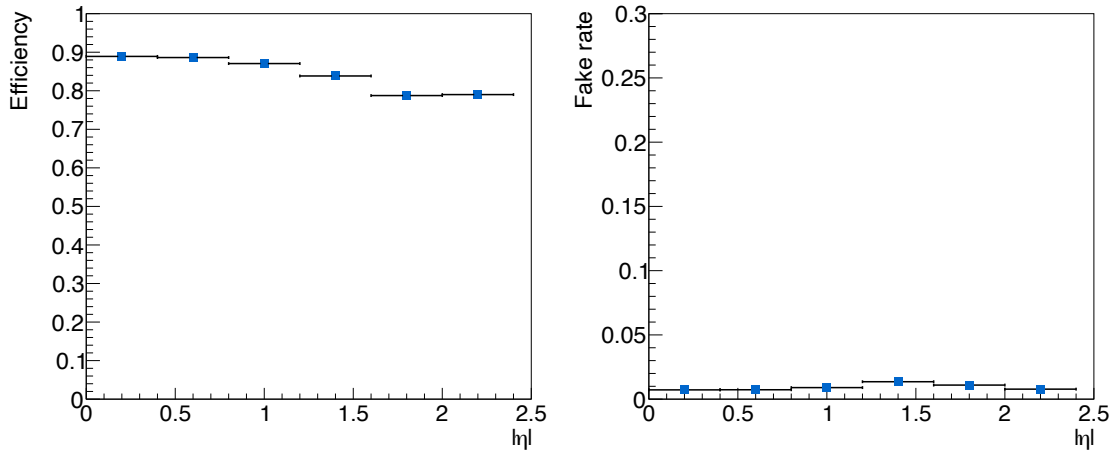
4.4 ATLAS Track Reconstruction Performance

The performance of the track reconstruction algorithms in ATLAS has great impact on the performances of all downstream algorithms that make use of reconstructed charged particle trajectories as part of their input, such as algorithms for the reconstruction of interaction vertices, as further discussed in Chapter 5 and beyond.
In the following, some tracking performance quantities are therefore briefly discussed, especially those that are important in the context of vertex reconstruction.
Fig. 4.5 shows the average d_0 and z_0 impact parameter resolutions of all reconstructed charged particle trajectories in $t\bar{t}$ events with a $\langle\mu\rangle = 38$ profile. While a very high resolution below 50 µm can be seen for the transverse impact parameter resolution around $|\eta| \approx 0$, the best achieved average longitudinal resolution is about 100 µm. A strong degradation for forward tracks with high pseudo-rapidity can be observed in both cases.
The average track reconstruction efficiency, defined as the probability for a charged truth particle to be successfully reconstructed as a track in the ATLAS Inner Detector, of all reconstructed tracks in $\langle\mu\rangle = 38$ $t\bar{t}$ events as a function of the absolute track pseudo-rapidity value $|\eta|$ is shown in Fig. 4.6(a). A very high, close to 90% efficiency is observed for central tracks while reconstruction efficiencies of about 80% are seen in the more forward region.
The average track reconstruction fake rate, defined as the probability of a reconstructed track to have the majority of its associated hits originating from multiple different truth particles, and therefore not representing a real charged particle trajectory, is shown in Fig. 4.6(b). The fake rate stays well below $1 - 2\%$ across the full depicted pseudo-rapidity range with a slight increase towards higher absolute pseudo-rapidity values.
In order to keep the influence of badly reconstructed tracks low, many downstream algorithm usually apply dedicated selection cuts on their input track collections. More details on track quality selection in the context of primary vertex reconstruction will be given in Section 5.2.1.



(a) (b)

Figure 4.5: Average (a) d_0 and (b) z_0 impact parameter resolutions of all reconstructed charged particle trajectories in $\langle\mu\rangle = 38$ $t\bar{t}$ events as a function of the absolute track pseudo-rapidity value $|\eta|$. Note that the observed resolution degradation in the forward region is mainly driven by increased material interactions.



(a) (b)

Figure 4.6: (a) Average track reconstruction efficiency of all reconstructed tracks in $\langle\mu\rangle = 38$ $t\bar{t}$ events as a function of the absolute track pseudo-rapidity value $|\eta|$. A track is considered successfully reconstructed, if at least 50% of its hit content matches the true hits produced by a single truth particle. (b) Average track reconstruction fake rate of all reconstructed tracks in $\langle\mu\rangle = 38$ $t\bar{t}$ events as a function of the absolute track pseudo-rapidity value $|\eta|$. A reconstructed track is labeled as fake if the majority of hits associated to the track is not originating from a single particle. Note that the observed efficiency degradation in the forward region is mainly driven by increased material interactions.

Chapter 5

Reconstruction of Interaction Vertices

5.1 Introduction

The efficient and precise reconstruction of interaction vertices (usually referred to as *vertexing*) is essential for understanding the full kinematics of the physics processes happening during a collision event in a particle collider and is therefore of great importance in many physics analyses.

Two different types of interaction vertices need to be distinguished. As illustrated in Fig. 5.1, the crossing of two proton bunches (or bunches of other particles) leads to primary interactions of the colliding particles along the beam line. These primary interaction points are called *primary vertices*. A primary vertex can be classified as a *signal* vertex (also called *hard-scatter* vertex), signifying the primary collision with an interesting physics process that caused the event to be selected by the trigger system, or as a *pile-up* vertex. As already discussed in Section 3.1.2, pile-up vertices are additional primary minimum-bias collisions that are superimposed to the physics process of interest and are generally caused by high instantaneous luminosities.

The second type of interaction vertices are *secondary vertices*, which arise from photon conversions, cascade decays, $V0$ decays or decays of other long-lived particles, such as B-hadrons. Secondary vertices are typically significantly displaced from primary vertices ($\mathcal{O}$(mm) for the flight distance before the decay of a B-hadron or e.g. $\mathcal{O}$(cm) for photon

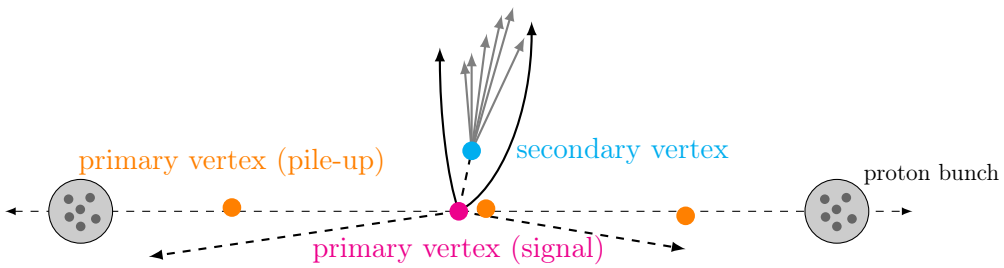


Figure 5.1: Illustration of different types of interaction vertices after the crossing of two proton bunches. Pile-up tracks are not depicted for the sake of simplicity.

conversions within the detector material) and can therefore be reconstructed separately from the primary interaction vertices.
Since the main work of this thesis will deal with the development and software implementation of *primary* vertex reconstruction algorithms, this chapter will primarily focus on the reconstruction of primary vertices, whose knowledge is inevitable for many analyses and downstream reconstruction algorithms, such as b-tagging, jet energy correction or pile-up suppression methods.
The reconstruction of primary vertices is based on an input collection of charged particle trajectories, reconstructed as described in Chapter 4, and can typically be subdivided into two separate stages:

- **Vertex finding** algorithms (also called *vertex finders*) form groups of tracks that are thought to have originated from a common interaction vertex and therefore deal with the association of tracks to vertices. The two main goals of a primary vertex finder are to accurately separate tracks originating from secondary vertices from those belonging to the primary ones and, at the same time, to properly associate primary tracks to the correct primary vertex, a task that can become extremely challenging if many close-by primary vertices are present in e.g. high pile-up environments.
- **Vertex fitting** algorithms (also called *vertex fitters*) are pure mathematical tools that estimate a common intersection point (i.e. the vertex position) and its uncertainties, given a set of tracks that are thought to belong to a single vertex. Since vertex fitters do usually not care whether the hypothesis that the set of given tracks does indeed intersect at a common vertex is true or not, the correct formation of track clusters performed by the vertex finder is crucial.

Additionally, vertex *seed finding* algorithms are usually employed during the vertex finding procedure to retrieve a first vertex estimate, a so-called vertex seed. Vertex seed finders therefore form another important building block of vertex reconstruction, as for most primary vertexing approaches a vertex will be lost if no corresponding accurate vertex seed could be found.
Two of the most widely used approaches to primary vertex reconstruction are the so-called *fitting-after-finding* and *finding-through-fitting* approaches. While the first one iteratively fits individual vertices after groups of compatible tracks have been formed from a seed track collection and is therefore based on a clear separation between the fitting and finding stage, the latter approach implements a strong interplay between the vertex finder and fitter stages that allows for simultaneous fits of multiple vertices, which is particularly favorable for environments with high vertex densities.
With the knowledge of the fitted vertex positions, the track parameters are usually also updated by constraining the tracks to pass through their associated vertex.
The reconstructed primary vertices of multiple dedicated events can be used to determine the interaction region, called *beam spot*, where the collisions within the detector take place. The exact size of the beam spot depends on the chosen operating parameters of the collider, but in general it follows a three-dimensional Gaussian distribution with an approximate

transverse size of $\sigma_{x,y} \approx 15\,\mu$m and longitudinal size of $\sigma_z \approx 55\,$mm [101] in case of the ATLAS experiment at the LHC. The exact knowledge of the beam spot position can also be used to constrain the vertex fits in the transverse plane, whereas the z-positions of the vertices are essentially unconstrained, given the large longitudinal size of the primary interaction region, and need to be precisely reconstructed by the primary vertex finder algorithms.
This chapter will exclusively cover algorithmic and mathematical aspects of vertex reconstruction. All of the presented algorithms and developments for primary vertex reconstruction have been implemented in an entirely new vertex reconstruction software suite that has been developed in the course of this thesis. Software implementation details as well as CPU and physics performance studies will be presented in the context of their actual implementation in Chapter 6 and Chapter 7, respectively.

5.2 Primary Vertex Finding

5.2.1 Track Selection

The input to a primary vertex finder algorithm is a subset of all charged particle trajectories, reconstructed in the ATLAS Inner Detector as described in Chapter 4. Among all reconstructed tracks, only those tracks are selected that pass a set of tight selection requirements in order to ensure a low rate of fake tracks (i.e. tracks formed from a random combination of detector hits, not belonging to a real particle) and to reduce the contamination from tracks originating from secondary interactions.
The following primary vertexing track selection criteria are applied (ATLAS baseline from data taking in Run 2 [102]):

- $p_T > 400\,$MeV
- $|d_0| < 4\,$mm, $\sigma(d_0) < 5\,$mm, $\sigma(z_0) < 10\,$mm
- Number of SCT hits ≥ 4
- Number of silicon (SCT or Pixel) hits ≥ 9, no pixel holes
- For $|\eta| \geq 1.65$: number of silicon hits ≥ 11, less than two SCT holes
- Number of IBL hits + number of B-layer (closest pixel layer) hits ≥ 1
- Not more than one shared pixel hit or two shared SCT hits

As the track p_T requirement for the selection of reconstructed tracks has been raised to $p_T > 500\,$MeV for Run 3 in order to reduce the average event size on disk, the corresponding cut in the primary vertexing track selection has been adapted as well to $p_T > 500\,$MeV for consistency and reproducibility [103].
After the set of reconstructed tracks has been selected based on the above defined requirements, primary vertices can be reconstructed as explained in the following sections.

5.2.2 Iterative Vertex Finder Algorithm

Originally developed for ATLAS primary vertex reconstruction during LHC Run 1, the *Iterative Vertex Finder* (IVF) [103] represents an algorithmic manifestation of a fitting-after-finding approach to primary vertexing. Finding and fitting stages are completely decoupled with vertices being iteratively found and afterwards fitted while tracks that have already been used for the vertex fit are removed from the track collection.
Fig. 5.2 diagrammatically illustrates the IVF algorithm, which starts with assigning all input tracks that passed dedicated selection cuts for excluding fake, badly measured or secondary tracks to the track collection for vertex seed finding.
Every vertex finder iteration now begins with the vertex finding step, forming groups of tracks that are expected to have originated from a common vertex. For this, the *vertex seed finder* (see Section 5.4) is invoked, returning a first vertex position estimate, the so-called *vertex seed*. A set of nearby tracks that are compatible with possibly originating from the given seed position are selected based on their closest approach distance to the vertex estimate. In ATLAS, compatible tracks are defined as tracks whose impact parameter points (d_0, z_0) are within a distance

$$d = \lambda \sqrt{\sigma^2(d_0) + \sigma^2(z_0)} \tag{5.1}$$

to the given vertex seed position, where λ denotes a tunable configuration parameter of the algorithm, set to $\lambda = 12$ for ATLAS, and $\sigma^2(d_0)$ as well as $\sigma^2(z_0)$ are the transverse and longitudinal impact parameter uncertainties, respectively.
Subsequently, in the vertex fitting step, the linearized track parameters of the selected tracks are passed to the vertex fitter (see Section 5.3) to determine the exact vertex position and its uncertainties. The vertex fitter can make use of additional information such as the beam spot, which can be seen as an extra three-dimensional Gaussian-distributed measurement in the fit that further constraints the vertex position in the x- and y-direction. Especially secondary tracks, whose track parameters are in general not compatible with the beam spot size in the transverse plane, can therefore easily be regarded as outliers and consequently down-weighted or discarded during the vertex fit. In addition to the beam spot constraint in the transverse plane, some vertex seed finders estimate the width of the vertex seed in z-direction and can therefore provide an additional constraint along the z-axis to the vertex fitter (see Section 5.4.2, 5.4.3 and 5.4.4.) After the vertex fit, the fitter can update, if desired, the track parameters with the newly acquired knowledge of the exact vertex position.
All tracks that significantly contributed to the vertex fit, i.e. their weight in the vertex fit is above a certain threshold (which is chosen to be 0.01 in ATLAS), are now removed from the track collection and cannot be used in subsequent iterations. The vertex finding procedure continues iteratively with the remaining tracks until one of two abort criteria, either less than two tracks remain for seeding in the seed track collection or the maximum number of iterations $N_{\text{max}}^{\text{iter}}$ (set to $N_{\text{max}}^{\text{iter}} = 200$ in ATLAS), is met.
Since tracks are immediately removed from the track pool after they have been used for

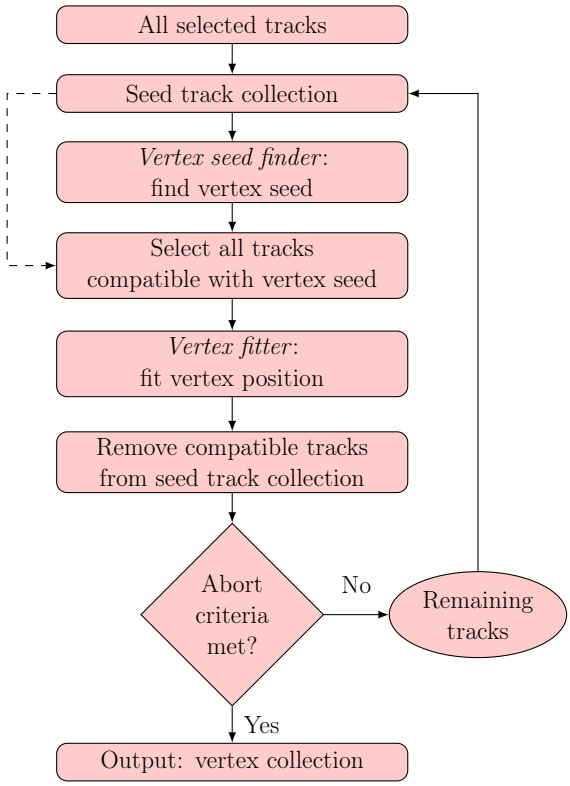


Figure 5.2: Diagrammatic representation of the Iterative Vertex Finder algorithm.

a vertex fit, single tracks are always only associated to at most one vertex at a time, as opposed to the *Adaptive Multi-Vertex Finder* described in the next section, where a single track can simultaneously be associated to several vertices during the fit.

5.2.3 Adaptive Multi-Vertex Finder Algorithm

As opposed to the IVF algorithm, which is based on a strict separation of the vertex finding and vertex fitting processes, the *Adaptive Multi-Vertex Finder* (AMVF) [103] implements a strong interplay between the finding and fitting stages in a so-called *finding-through-fitting* approach. Single tracks can be associated to several vertices simultaneously and vertex positions are determined in multi-vertex fits using deterministic annealing schemes.
While the outline of the AMVF algorithm, as diagrammatically depicted in Fig. 5.3, looks similar to the one of the IVF, the algorithmic procedure exhibits some key difference that will be discussed in the following. As for the IVF, tracks that are not yet associated to a vertex will be considered in the seed track collection and passed to the vertex seed finder to determine a first vertex estimate at the beginning of every AMVF iteration. The default seed finder for the AMVF is the *Gaussian Track Density Seed Finder* (see Section 5.4.2) as it provides, in addition to the vertex seed position, information about the longitudinal width of the seed, which can be used to further constrain the multi-vertex fit in z-direction and hence immensely increase the algorithm's performance, as further detailed in Section 5.3.3.

Tracks that are compatible with the current vertex candidate are selected in the next step. A fundamental difference to the IVF algorithm is that the AMVF considers all tracks having passed the quality criteria selection, and thus not only unassigned ones, for being potentially compatible with the vertex seed. Therefore, also tracks that are already assigned to one or more previously fitted and accepted vertices can be assigned to the new vertex candidate, allowing the simultaneous association of single tracks to multiple vertices. The track-to-vertex compatibility is determined by a cut on the impact parameter significance

$$s = \sqrt{\left(\frac{d_0}{\sigma(d_0)}\right)^2 + \left(\frac{z_0}{\sigma(z_0)}\right)^2}, \tag{5.2}$$

taking the track's distance to the vertex candidate as well as the impact parameter uncertainties into account. In ATLAS, a significance cut of $s < 5$ is chosen for the AMVF.

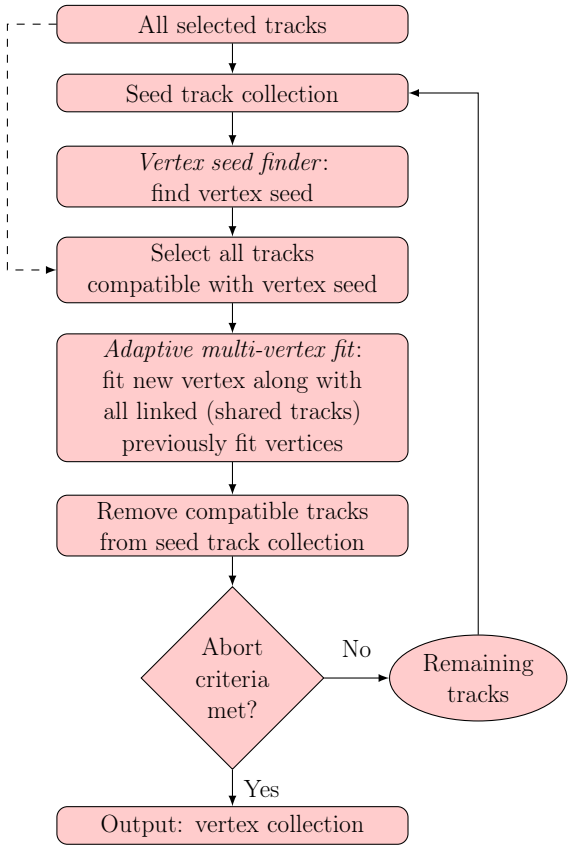


Figure 5.3: Diagrammatic representation of the Adaptive Multi-Vertex Finder algorithm.

In the subsequent vertex fitting step, a so-called *adaptive multi-vertex fit* (as discussed in detail in Section 5.3.3) of the current vertex candidate V_1 together with all affected previously found vertices, i.e. vertices that are linked to V_1 through an arbitrarily long chain of shared tracks, will be performed. If, for instance, a track T_1 is compatible with V_1 and also already attached to a previously found vertex V_2 and V_2 in turn shares tracks with previously found vertices V_3 and V_4, a simultaneous (re-)fit of all four vertices $V_1, \ldots, V_4$ will be triggered in the current iteration. The multi-vertex fit forces vertices to compete for

shared tracks with the track weights being adaptively determined based on their compatibility with the respective vertex using a deterministic annealing approach. Additionally, the fit is subject to a longitudinal and transverse constraint as given by the seed and beam spot position, respectively.
The fitted vertex is accepted if it has at least two compatible tracks from the seed track collection that were not already used in any other previous vertex fit. A vertex is therefore not accepted if it exclusively contains shared tracks. Additionally, the new vertex must be outside of a significance region of 3σ to any other previously fitted vertex. All compatible tracks are then removed from the seed track collection and the vertex finding procedure continues iteratively with the remaining tracks. If less than two tracks remain in the seed track collection or the maximum number of AMVF iterations (set to $N_{\text{max}}^{\text{iter}} = 100$ in ATLAS) is reached, the algorithm terminates and outputs the vertex collection.

5.2.4 Hard-Scatter Selection

Primary vertex finder algorithms do in general *not* distinguish between hard-scatter and pile-up vertices. Only after the collection of all reconstructed primary vertices is returned by the algorithm, the hard-scatter vertex has to be correctly identified among the many additional pile-up vertices.
As most of the interesting hard-scatter processes usually exhibit a harder transverse momentum spectrum of contributing tracks compared to pile-up interactions, an effective hard-scatter identification is often guaranteed by selecting the primary vertex with the highest sum of squared transverse momenta of contributing tracks: $\max\left(\sum p_T^2\right)$.
Specific analyses may choose to overwrite this selection criterion and use a more effective method for their desired signal process selection, as e.g. done in the case of $H \to \gamma\gamma$ signal events (see e.g. Ref. [104]).

5.2.5 Classification of Reconstructed Vertices

The main goal of a primary vertex finder algorithm in high pile-up environments is to find and reconstruct as many of the primary interaction vertices as possible, especially to ensure a clean and high-quality reconstruction of the hard-scatter interaction with as little pile-up contamination as possible.
In order to be able to quantify the performance, reconstructed vertices and charged particle tracks from simulated data need to be compared to the true underlying primary pp interactions and generated particles in the Monte Carlo event. Reconstructed tracks can be associated to a simulated pp interaction with the help of their respective truth particle, which is determined based on the number of matching simulated detector hits (see Ref. [101]). Using the stored Monte Carlo particle history, the matched truth particle can be traced back to the simulated pp interaction it originated from and therefore create the link between a reconstructed track and a true vertex.
By doing so, each reconstructed vertex can be assigned to one of the following classes, as defined and used in ATLAS (definitions taken from Ref. [103]):

- CLEAN/MATCHED: At least 70% of the total track weight in the reconstructed vertex originates from a single simulated *pp* interaction.
- MERGED: Less than 70% of the total track weight in the reconstructed vertex originates from any single simulated *pp* interaction. Hence, two or more simulated interactions contribute significantly to the accumulated track weight.
- SPLIT: A single simulated *pp* interaction contributes the largest track weight to two or more reconstructed vertices. The reconstructed vertex with the largest track $\sum p_T^2$ is classed as either CLEAN or MERGED, whilst the other(s) are labeled SPLIT.
- FAKE: Fake tracks contribute more weight to the reconstructed vertex than any simulated *pp* interaction.

Fig. 5.4 shows the number of reconstructed vertices in simulated $t\bar{t}$ events as a function of the true number of *pp* interactions per bunch-crossing for the AMVF and IVF as well as the different categories the AMVF-reconstructed vertices can be classified in. The majority of vertices reconstructed by the AMVF is classed CLEAN/MATCHED, while a smaller but still significant amount of vertices contain large track contributions from two or more simulated interactions and are therefore classified as MERGED. The number of vertices labeled SPLIT and FAKE is extremely small and often negligible.
In general, the more sophisticated primary vertexing approach deployed by the AMVF helps to recover vertices that are lost in the IVF reconstruction.
The quality of the reconstructed hard-scatter vertex can also be used to classify the event as a whole. The definitions used in ATLAS (taken from Ref. [103]) are:

- CLEAN/MATCHED: The event contains a CLEAN reconstructed vertex originating from the true HS interaction, and the HS interaction does not contribute more than 50% of the accumulated track weight to any other vertex.
- LOWPU (low pile-up): The event contains a MERGED vertex with at least 50% of the accumulated track weight coming from the simulated HS interaction.
- HIGHPU (high pile-up): The event contains a MERGED vertex with its main contribution coming from a simulated pile-up interaction, and in which the HS interaction contributes between 1% and 50% of the accumulated track weight.
- PUREPU (pure pile-up): The event does not contain any reconstructed vertex with at least 1% accumulated track weight from the HS interaction.

5.3 Vertex Fitting

Given a set of tracks that are expected to have originated from a common interaction point, vertex fitting deals with the mathematical determination of the exact vertex position and its uncertainties. The fitted position can subsequently be used to reevaluate the input track parameters and refit their trajectories with the knowledge of their common intersection point. Several different method for vertex fitting are discussed in the following.

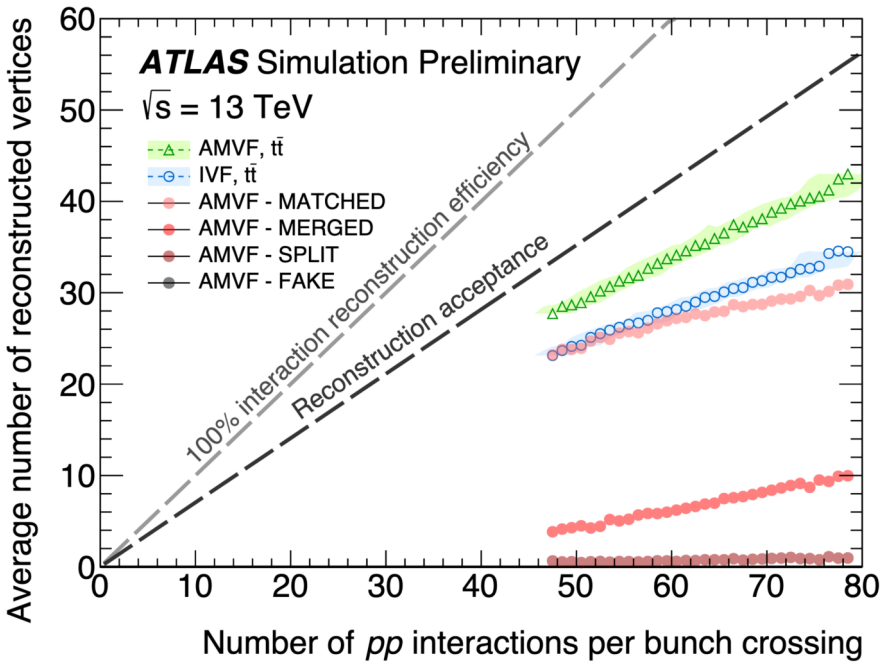


Figure 5.4: Average number of reconstructed vertices by the AMVF and IVF as a function of the number of true pp interactions per bunch-crossing in simulated $t\bar{t}$ events with $\sqrt{s} = 13\,\text{TeV}$. Additionally, the classification of the AMVF-reconstructed vertices in the classes MATCHED, MERGED, SPLIT and FAKE is shown. The grey dashed line illustrates a hypothetical perfect, 100% efficient primary vertex reconstruction, while the black dashed line gives an estimate of the highest possible reconstruction efficiency, taking the input track acceptance into account [103].

5.3.1 Billoir Method

The Billoir method [105] for vertex fitting uses a least-square approach to estimate the vertex position $\vec{r} = (x_v, y_v, z_v)$ and its uncertainties as well as the track momenta $\vec{p}_i$ for each track $i = 1, \ldots, N_{\text{trk}}$ at the vertex position. The input is a set of N_{trk} track measurements $\vec{q}_{\text{meas},i}$ and their covariance matrices $\mathbf{C}_i$, expressed in an arbitrary but fixed reference frame. The dimension of the measured parameters $\vec{q}_{\text{meas},i}$ depends on the chosen track parametrization, which, in the case of ATLAS, will be five perigee parameters as discussed in Section 4.3.

The least-square ansatz can be written as

$$\chi^2 = \chi^2(\vec{r}, \vec{p}_1, \ldots, \vec{p}_{N_{\text{trk}}}) = \sum_{i=1}^{N_{\text{trk}}} (\vec{q}_{\text{meas},i} - \vec{q}_{\text{pred},i}(\vec{r}, \vec{p}_i))^T \mathbf{C}_i^{-1} (\vec{q}_{\text{meas},i} - \vec{q}_{\text{pred},i}(\vec{r}, \vec{p}_i)), \quad (5.3)$$

where the functions $\vec{q}_{\text{pred},i}(\vec{r}, \vec{p}_i)$ denote predicted parameters a track i would have if it passes exactly through the vertex position $\vec{r}$ with momentum $\vec{p}_i$.

The goal of the vertex fit is now to find an optimal set of $\vec{r}$ and $\vec{p}_1, \ldots, \vec{p}_{N_{\text{trk}}}$ that minimizes

$\chi^2(\vec{r}, \vec{p}_1, \ldots, \vec{p}_{N_{\text{trk}}})$, i.e. the following set of equations

$$\begin{aligned} \frac{\partial \chi^2}{\partial \vec{r}} &= 0 \\ \frac{\partial \chi^2}{\partial \vec{p}_i} &= 0 \end{aligned} \tag{5.4}$$

needs to be solved. In order to get an analytic solution for Eq. (5.4), the functions $\vec{q}_{\text{pred},i}(\vec{r}, \vec{p}_i)$, denoted as $\vec{q}_i(\vec{r}, \vec{p}_i)$ for the sake of conciseness in the following, are linearized in a first order Taylor expansion around an initial estimate of the vertex position $\vec{r}_0$ and track momenta $\vec{p}_{0,i}$:

$$\begin{aligned} \vec{q}_i(\vec{r}_0 + \delta\vec{r}, \vec{p}_{0,i} + \delta\vec{p}_i) &= \underbrace{\vec{q}_i(\vec{r}_0, \vec{p}_{0,i})}_{\vec{q}_{0,i}} + \underbrace{\frac{\partial \vec{q}_i(\vec{r}, \vec{p}_i)}{\partial \vec{r}}}_{\mathbf{D}_i} \delta\vec{r} + \underbrace{\frac{\partial \vec{q}_i(\vec{r}, \vec{p}_i)}{\partial \vec{p}_i}}_{\mathbf{E}_i} \delta\vec{p}_i \\ &= \vec{q}_{0,i} + \mathbf{D}_i \delta\vec{r} + \mathbf{E}_i \delta\vec{p}_i \end{aligned} \tag{5.5}$$

Here, $\delta\vec{r}$ and $\delta\vec{p}_i$ denote small variations around the linearization point, $\mathbf{D}_i$ and $\mathbf{E}_i$ are the Jacobians for the transformations from the position space $\vec{r}$ and the momentum space $\vec{p}$ to the track parameter space $\vec{q}$, respectively, evaluated for a track i at $(\vec{r}, \vec{p}_i) = (\vec{r}_0, \vec{p}_{0,i})$. With $\vec{r} = \vec{r}_0 + \delta\vec{r}$ and $\vec{p}_i = \vec{p}_{0,i} + \delta\vec{p}_i$, Eq. (5.5) can also be rewritten as

$$\begin{aligned} \vec{q}_i(\vec{r}, \vec{p}_i) &= \vec{q}_{0,i} + \mathbf{D}_i(\vec{r} - \vec{r}_0) + \mathbf{E}_i(\vec{p}_i - \vec{p}_{0,i}) \\ &= \mathbf{D}_i\vec{r} + \mathbf{E}_i\vec{p}_i + \underbrace{\vec{q}_{0,i} - \mathbf{D}_i\vec{r}_0 - \mathbf{E}_i\vec{p}_{0,i}}_{\vec{c}_{0,i}} \\ &= \mathbf{D}_i\vec{r} + \mathbf{E}_i\vec{p}_i + \vec{c}_{0,i} \end{aligned} \tag{5.6}$$

which is referred to as the *measurement equation*, as it directly relates the input measurements for the vertexing, i.e. the reconstructed track parameters $\vec{q}_{\text{meas},i}$, to the vertexing output fit variables, the vertex position $\vec{r}$ and the respective track momentum $\vec{p}_i$.
Using the linearized track parameters from Eq. (5.5) in the least-square ansatz, the resulting expression

$$\chi^2 = \sum_{i=1}^{N} (\delta\vec{q}_i - \mathbf{D}_i\delta\vec{r} - \mathbf{E}_i\delta\vec{p}_i)^T \mathbf{C}_i^{-1} (\delta\vec{q}_i - \mathbf{D}_i\delta\vec{r} - \mathbf{E}_i\delta\vec{p}_i) \tag{5.7}$$

with $\delta\vec{q}_i = \vec{q}_{\text{meas},i} - \vec{q}_{0,i}$ can now be used to solve Eq. (5.4).
As derived in Ref. [105], the solution for the vertex position and the updated track momenta at the vertex results in

$$\vec{r} = \vec{r}_0 + \delta\vec{r} = \vec{r}_0 + \left(\mathbf{A} - \sum_{i=1}^{N} \mathbf{B}_i \mathbf{G}_i^{-1} \mathbf{B}_i^T \right)^{-1} \left(\vec{T} - \sum_{i=1}^{N} (\mathbf{B}_i \mathbf{G}_i^{-1})^T \vec{U}_i \right) \tag{5.8}$$

$$\vec{p}_i = \vec{p}_{0,i} + \delta\vec{p}_i = \vec{p}_{0,i} + \mathbf{G}_i^{-1}(\vec{U}_i - \mathbf{B}_i^T \delta\vec{r}) \tag{5.9}$$

with vertex position and track momentum covariance matrices

$$\mathbf{C}_V = \left(\mathbf{A} - \sum_{i=1}^{N} \mathbf{B}_i \mathbf{G}_i^{-1} \mathbf{B}_i^T \right)^{-1} \tag{5.10}$$

$$\mathbf{C}_p = \mathbf{G}_i^{-1} + (\mathbf{B}_i \mathbf{G}_i^{-1})^T \mathbf{C}_V (\mathbf{B}_i \mathbf{G}_i^{-1}), \tag{5.11}$$

where the following abbreviations have been introduced:

$$\begin{aligned} \mathbf{A} &= \sum_{i=1}^{N} \mathbf{D}_i^T \mathbf{C}_i^{-1} \mathbf{D}_i, \qquad & \mathbf{B}_i &= \mathbf{D}_i^T \mathbf{C}_i^{-1} \mathbf{E}_i \\ \vec{T} &= \sum_{i=1}^{N} \mathbf{D}_i^T \mathbf{C}_i^{-1} \delta\vec{q}_i, \qquad & \vec{U}_i &= \mathbf{E}_i^T \mathbf{C}_i^{-1} \delta\vec{q}_i \\ \mathbf{G}_i &= \mathbf{E}_i^T \mathbf{C}_i^{-1} \mathbf{E}_i \end{aligned} \tag{5.12}$$

In practice, the Billoir vertex fitting procedure is employed in an iterative manner, where the previously fitted vertex position is used as a starting point for the next iteration, assuming that the track parameter approximation is becoming increasingly more accurate as the fitted position is successively getting closer to the true vertex.

5.3.2 Kalman Filter

The Kalman filter formalism presented in Section 4.2.3 can also be applied to vertex fitting [93, 106]. Here, the state vector $\vec{x}$ consists of the vertex position $\vec{r}$ to be determined and at every step k, when a new track (regarded as a virtual measurement for the vertex fit) is added to the system, the state is augmented with the momentum vector $\vec{p}_k$ of the newly added track.

At every step k, the Kalman filter provides an estimation of the vertex position and its associated covariance matrix, taking all k tracks into account that have been added until the current step. While the vertex position estimation is considered completed after the filtering stage when all available tracks are included in the fit, the individual track momenta $\vec{p}_k$ have only been updated once at their respective step k, not taking the vertex position updates at all subsequent steps into account. Therefore, the final vertex position result is back-propagated through all previous Kalman steps during the smoothing stage, in order to obtain a complete estimate of all updated track momenta.

Since neither a propagation needs to be performed nor a noise term needs to be added when evolving from step $k-1$ to step k, the Kalman evolution equation for vertex fitting becomes trivial and can be written as

$$\vec{x}_k = \vec{x}_{k-1}. \tag{5.13}$$

A linear dependency between the measurement $\vec{m}_k$, i.e. the reconstructed track parameters $\vec{q}_k$, and the vertex position as well as the respective track momentum is given by the linearized measurement equation

$$\vec{m}_k = \mathbf{H}_k \vec{x}_k + \vec{\epsilon}_k = \mathbf{D}_k \vec{r} + \mathbf{E}_k \vec{p}_k + \vec{c}_{0,k}, \tag{5.14}$$

which was already presented in Eq. (5.6).
The updated state vector after adding a new track at step k is again determined by a Kalman filter step.
In the Kalman gain formalism, as already presented in the context of track fitting, the new state vector $\vec{x}_k$ and its associated covariance matrix $\mathbf{C}_k$ are obtained by applying the Kalman gain matrix

$$\mathbf{K}_k = \mathbf{C}_k^{k-1} \mathbf{H}_k^{\mathrm{T}} \left[\mathbf{V}_k + \mathbf{H}_k \mathbf{C}_k^{k-1} \mathbf{H}_k^{\mathrm{T}} \right]^{-1} \tag{5.15}$$

to the measurement residual and updating the state accordingly:

$$\begin{aligned} \vec{x}_k &= \vec{x}_k^{k-1} + \mathbf{K}_k \left(\vec{m}_k - \mathbf{H}_k \vec{x}_k^{k-1} \right) \\ \mathbf{C}_k &= (\mathbb{1} - \mathbf{K}_k \mathbf{H}_k) \mathbf{C}_k^{k-1} \end{aligned} \tag{5.16}$$

An alternative approach, which is mathematically equivalent to the gain formalism, is the so-called Kalman *weighted mean* formalism, in which the new state $\vec{x}_k$ is determined by a combination of the previous state $\vec{x}_{k-1}$ and information from the new measurement to be added, weighted by the inverse state covariance matrix $\left(\mathbf{C}_k^{k-1} \right)^{-1}$ and inverse covariance matrix of the new measurement $\mathbf{V}_k^{-1}$:

$$\begin{aligned} \vec{x}_k &= \mathbf{C}_k \left[\left(\mathbf{C}_k^{k-1} \right)^{-1} \vec{x}_k^{k-1} + \mathbf{H}_k^{\mathrm{T}} \mathbf{V}_k^{-1} \mathbf{H}_k \right] \\ \mathbf{C}_k &= \left[\left(\mathbf{C}_k^{k-1} \right)^{-1} + \mathbf{H}_k^{\mathrm{T}} \mathbf{V}_k^{-1} \vec{m}_k \right]^{-1} \end{aligned} \tag{5.17}$$

While both of the above approaches will yield the exact same results, they differ in the type of application they are best suited for. The gain formalism requires an inversion of a matrix of the same dimension as the measurement vector, whereas the weighted mean formalism inverts a matrix of the state vector dimension. In case the measurement dimension is smaller than the state dimension, the gain formalism is therefore usually faster as a smaller matrix needs to be inverted.
Additionally, since the weighted mean formalism is expressed in terms of weight matrices (as opposed to covariance matrices in the gain formalism), very large uncertainties can be assigned to the track momenta in the initial state vector, a feature that is often required as initial track momentum estimates are usually not available and their initial uncertainties are therefore set to infinity. If, on the other hand, the application of an exact constraint with zero uncertainty is desired, the Kalman gain formalism is favored.
The fitted vertex position at step k in the Kalman weighted mean formalism is now given

by

$$\vec{r}_k = \mathbf{cov}(\vec{r})_k \left[\mathbf{cov}(\vec{r})_k^{k-1} \vec{r}_k^{k-1} + \mathbf{D}_k^{\mathrm{T}} \tilde{\mathbf{C}}(\vec{q})_k^{-1} (\vec{q}_{\mathrm{meas}} - \vec{c}_{0,k}) \right] \tag{5.18}$$

with its associated covariance matrix

$$\mathbf{cov}(\vec{r})_k = \left[\left(\mathbf{cov}(\vec{r})_k^{k-1} \right)^{-1} + \mathbf{D}_k^{\mathrm{T}} \tilde{\mathbf{C}}(\vec{q})_k^{-1} \mathbf{D}_k \right]^{-1}, \tag{5.19}$$

where the abbreviation

$$\tilde{\mathbf{C}}(\vec{q})_k^{-1} = \mathbf{cov}(\vec{q})_k^{-1} \left[\mathbb{1} - \mathbf{E}_k \left[\mathbf{E}_k^{\mathrm{T}} \mathbf{cov}(\vec{q})_k^{-1} \mathbf{E}_k \right]^{-1} \mathbf{E}_k^{\mathrm{T}} \mathbf{cov}(\vec{q})_k^{-1} \right] \tag{5.20}$$

has been introduced for simplicity.
Furthermore, as shown in Refs. [93, 106], the solutions of the weighted mean Kalman formalism and the iterative application of the Billoir fitter are mathematically equivalent, while the Kalman filter procedure can reduce the number of required covariance matrix inversions by a factor of N_{trk}.

5.3.3 Adaptive Multi-Vertex Fitter

Most vertex fitting methods rely on the minimization of track parameter residuals between the measured and predicted track parameters with respect to a vertex position $\vec{r}$. As already discussed in Section 5.3.1, a corresponding χ^2 term can be written as

$$\chi^2 = \sum_{i=1}^{N} \Delta\vec{q}_i^{\,T} \mathbf{C}_i^{-1} \Delta\vec{q}_i, \tag{5.21}$$

where $\Delta\vec{q}_i = \vec{q}_{\mathrm{meas},i} - \vec{q}_{\mathrm{pred},i}(\vec{r})$ denote the track parameter residuals for all tracks $i = 1, \ldots, N$ to be minimized.
Since the χ^2 value depends quadratically on the track residuals, it is extremely sensitive to outlying measurements, so-called *outliers*, that can significantly bias the outcome of the fit. In the context of vertex reconstruction, these outliers can either be

- reconstructed tracks that originated from a different, close-by vertex and have been incorrectly assigned to the current vertex fit by the vertex finder, or
- fake reconstructed tracks that do not represent a real particle trajectory but were formed from a random combination of hits in the detector.

Reducing the influence of such outlying track measurements is therefore an important task that needs to be solved during the reconstruction of interaction vertices. One possibility to address this issue has already been discussed in Section 5.2.2 and 5.2.3, where the vertex finder algorithms include or exclude tracks for the subsequent vertex fit based on their compatibility with the current vertex estimate and therefore try to reduce the influence of outliers as much as possible.
Another approach is to deal with outlying measurements directly during an iterative vertex

fit by modifying the χ^2 function in such way that non-compatible tracks are *adaptively* down-weighted and therefore contribute less to the fit, as explained in the following [106].

Adaptive Vertex Fitting

A modified χ^2 function that takes a track's compatibility with the current vertex into account can be defined as

$$\chi^2 = \sum_{i=1}^{N} \omega(\hat{\chi}_i^2) \Delta\vec{q}_i^{\,T} \mathbf{C}_i^{-1} \Delta\vec{q}_i, \tag{5.22}$$

where $\omega(\hat{\chi}_i^2)$ denotes a weight function for each track i, which depends on the compatibility of the track with respect to the vertex of interest. This compatibility value is usually computed as the track residual contribution to the overall χ^2 value with respect to the last best vertex position estimate (typically given by the last fitted vertex position in an iterative χ^2-minimization approach) and is denoted as $\hat{\chi}_i^2$. Tracks that have a low probability of being compatible with the vertex position in the previous iteration, and therefore exhibit a large $\hat{\chi}_i^2$-distance to that vertex position, are down-weighted and contribute less to the overall χ^2 calculation in Eq. (5.22).

A major drawback of this approach lies in the huge impact outliers can have during the very first iterations: the initially fitted vertex position usually shows large contributions from outliers that could not be down-weighted yet. The vertex position, with respect to which the track weighting factors $\omega(\hat{\chi}^2)$ are computed in the subsequent iteration, can therefore be largely biased toward outlying measurements from the very beginning and tracks that would normally be perfectly compatible with the current vertex of interest are down-weighted or even excluded from the fit. Additionally, outliers can be assigned large weights as the initial vertex position estimate was largely biased by their presence.

A method that was developed to overcome these limitations is the *Adaptive Vertex Fitter* [107], which introduces a temperature parameter T to the weight function that can be used for simulated annealing of the system, as explained in the following.

The weight function is defined to be dependent on the track's $\hat{\chi}^2$-distance to the vertex as well as the newly introduced temperature T according to

$$\omega(\hat{\chi}_i^2, T) = \frac{\exp\left(-\frac{\hat{\chi}_i^2}{2T}\right)}{\exp\left(-\frac{\hat{\chi}_{\text{cut}}^2}{2T}\right) + \exp\left(-\frac{\hat{\chi}_i^2}{2T}\right)}, \tag{5.23}$$

representing a Fermi function, where the point of transition between large and small track weights is governed by the cut-off value $\hat{\chi}_{\text{cut}}^2$.

As shown in Fig. 5.5(a), higher system temperatures T lead to a more smoothed-out weight function, while a cooled-down system with low temperatures results in a steeper weight function around the cut-off value. In the context of compatibility weights being assigned to tracks during a vertex fit, this means that for high temperature values only a slight weight preference is given to tracks with small $\hat{\chi}^2$-distances with respect to the last vertex position, whereas a low system temperature causes tracks to be drastically up-weighted for $\hat{\chi}^2 < \hat{\chi}_{\text{cut}}^2$ or down-weighted for $\hat{\chi}^2 > \hat{\chi}_{\text{cut}}^2$.

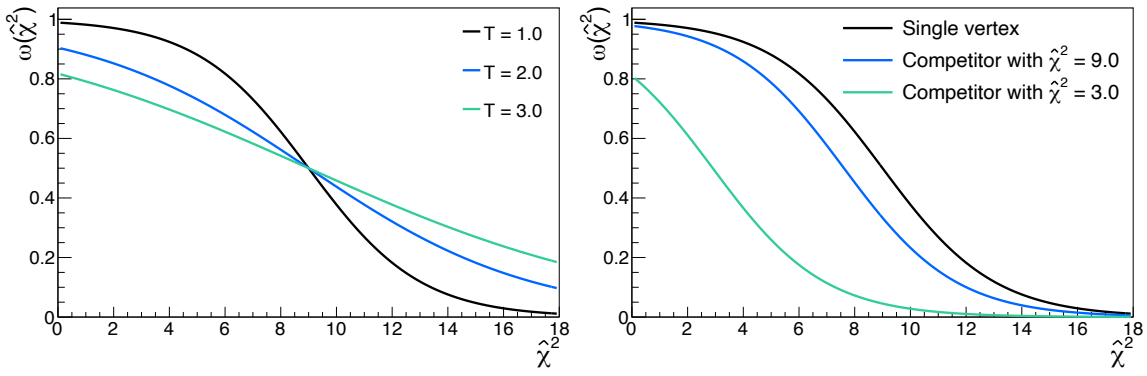
A temperature decay can easily be deployed by a deterministic annealing scheme with a set of fixed, over time decreasing temperatures such that outlying measurements contribute only little at the very beginning of the vertex fit and will eventually be discarded as the temperature declines and the weight assignment becomes stricter.

Adaptive Multi-Vertex Fitting

An extension of the Adaptive Vertex Fitter was presented in Ref. [108] as the *Adaptive Multi-Vertex Fitter* that allows the simultaneous fit of N tracks to M vertices and therefore the competition between vertices for tracks as single tracks can be associated to several vertices at once. The weight of track i with respect to vertex j is given by an extended version of the weight function from Eq. (5.23) as

$$\omega_{ij}(\hat{\chi}^2_{i1}, \ldots, \hat{\chi}^2_{iM}, T) = \frac{\exp\left(-\frac{\hat{\chi}^2_{ij}}{2T}\right)}{\exp\left(-\frac{\hat{\chi}^2_{\text{cut}}}{2T}\right) + \sum_{k=1}^{M} \exp\left(-\frac{\hat{\chi}^2_{ik}}{2T}\right)}, \tag{5.24}$$

where the $\hat{\chi}^2$-distances between track i and all $k = 1, \ldots, M$ considered vertices, denoted as $\hat{\chi}^2_{ik}$, are taken into account. High compatibilities of a track to other vertices will accordingly increase the weight function's denominator and therefore down-weight the track with respect to the vertex of interest. This behavior is illustrated in Fig. 5.5(b) for three different cases with a fixed temperature value of $T = 1$.



(a) (b)

Figure 5.5: (a) Track weight with respect to a single vertex as a function of $\hat{\chi}^2$ for different temperatures T and a cut-off value of $\hat{\chi}^2_{\text{cut}} = 9$. (b) Track weight as a function of $\hat{\chi}^2$ for the case of no competing vertex as well as one competing vertex for different compatibility values with a cut-off at $\hat{\chi}^2_{\text{cut}} = 9$ and a fixed temperature $T = 1$.

In case of a single vertex, and therefore no other present competing vertices, the weight function just reduces to the one introduced for the Adaptive Vertex Fitter in Eq. (5.23) as can be seen by comparing the single-vertex graph of Fig. 5.5(b) to Fig. 5.5(a) for $T = 1$. If the considered track, however, is also compatible to a second vertex, the same $\hat{\chi}^2$-distance

to the vertex of interest will now lead to a smaller track weight compared to the single-vertex case. The higher the compatibility with other vertices, the harder it gets to score a large track weight and very small $\hat{\chi}^2$-distances are needed in order to reach a track weight close to 1.
The Adaptive Multi-Vertex Fitting method has proven to be of particular benefit in high track-multiplicity environments, where an efficient rejection of outliers, especially those stemming from incorrectly assigned tracks from neighboring vertices, is needed.

5.3.4 Helical Track Parameter Linearization

The most important input to the vertex fit is the set of track parameters, all of which need to be linearized in the vicinity of the estimated vertex position [105]. According to Eq. (5.6), the measurement equation linearized around a certain position $\vec{r}_0$ and point in momentum space $\vec{p}_0$ is given by

$$\vec{q}(\vec{r},\vec{p}) = \mathbf{D}\vec{r} + \mathbf{E}\vec{p} + \vec{c}_0 \tag{5.25}$$

with $\vec{r} = \vec{r}_0 + \delta\vec{r}$ and $\vec{p} = \vec{p}_0 + \delta\vec{p}$, where $\delta\vec{r}$ and $\delta\vec{p}$ denote small variations around the linearization point $(\vec{r}_0, \vec{p}_0)$.
In order to determine the linearized track parameters $\vec{q}(\vec{r},\vec{p})$, the position Jacobian $\mathbf{D}$ and momentum Jacobian $\mathbf{E}$ have to be evaluated.
An analytic computation of $\mathbf{D}$ and $\mathbf{E}$ was presented in Ref. [106], based on the assumption of an ideally helical track, following the equations of motion of a charged particle in a constant magnetic field. The track parametrization is given in the perigee frame, as illustrated in Fig. 5.6.
In addition to the five perigee track parameters $(d_0, z_0, \phi_P, \theta_P, (q/p)_P)$, the signed radius of the helix is defined as

$$\rho = \frac{\sin\theta}{q/p\, B_z}, \tag{5.26}$$

where $\theta = \theta_P = \text{const.}$ along the helix and B_z denotes the z-component of the constant magnetic field.

Additionally, a generic position V along the helix can be parametrized in terms of the trajectory parameters in point P and the reference point R with coordinates (x_R, y_R, z_R) as

$$\begin{aligned}
x_V &= x_R + d_0 \cos\left(\phi_P + \frac{\pi}{2}\right) + \rho\left[\cos\left(\phi_V + \frac{\pi}{2}\right) - \cos\left(\phi_P + \frac{\pi}{2}\right)\right] \\
y_V &= y_R + d_0 \sin\left(\phi_P + \frac{\pi}{2}\right) + \rho\left[\sin\left(\phi_V + \frac{\pi}{2}\right) - \sin\left(\phi_P + \frac{\pi}{2}\right)\right] \\
z_V &= z_R + z_0 - \frac{\rho}{\tan(\theta)}\left[\phi_V - \phi_P\right],
\end{aligned} \tag{5.27}$$

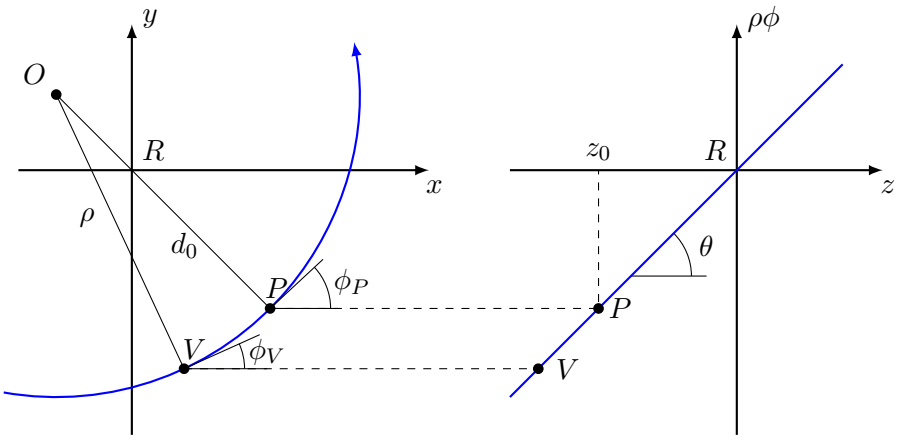


Figure 5.6: Illustration of a helical track trajectory in the transverse plane (left) and $\rho\phi - z$ plane (right). The perigee track parameters are defined in the point of closest approach P to the reference point R. A second point along the helix V is shown which is determined by its azimuthal angle ϕ_V.

where the exact position of V along the helix is specified by its azimuthal angle ϕ_V. In order to compute the position and momentum Jacobians of interest,

$$\mathbf{D} = \frac{\partial(d_0, z_0, \phi_P, \theta_P, (q/p)_P)}{\partial(x_V, y_V, z_V)} \tag{5.28}$$

$$\mathbf{E} = \frac{\partial(d_0, z_0, \phi_P, \theta_P, (q/p)_P)}{\partial(\phi_V, \theta_V, (q/p)_V)}, \tag{5.29}$$

indicating how the track perigee parameters in P change with variations of the generic position (x_V, y_V, z_V) and its momentum $(\phi_V, \theta_V, (q/p)_V))$, the perigee parameters $\vec{q} = (d_0, z_0, \phi_P, \theta_P, (q/p)_P)$ are expressed in terms of $(x_V, y_V, z_V, \phi_V, \theta_V, (q/p)_V))$ by exploiting the relations in Eq. (5.27) as:

$$\begin{aligned}
d_0 &= \rho + \mathtt{sgn}(d_0 - \rho)\sqrt{(x_V - x_R - \rho\sin(\phi_V))^2 + (y_V - y_R - \rho\cos(\phi_V))^2} \\
\phi_P &= \arctan\left(\frac{y_V - y_R - \rho\cos(\phi_V)}{x_V - x_R - \rho\sin(\phi_V)}\right) \\
z_0 &= z_V + z_R + \frac{\rho}{\tan(\theta)}\left[\phi_V - \phi_P\right] \\
(q/p)_P &= (q/p)_V \\
\theta_P &= \theta_V
\end{aligned} \tag{5.30}$$

Additionally, we find $\mathtt{sgn}(d_0 - \rho) \simeq -\mathtt{sgn}(\rho)$ with $|\rho| \gg |d_0|$ and define $h = \mathtt{sgn}(\rho)$ as well as $\Delta\phi = \phi_P - \phi_V$.

The required derivatives can now be computed and the analytic expressions for the Jacobians result in

$$\mathbf{D} = \begin{bmatrix} -h\frac{X}{S} & -h\frac{Y}{S} & 0 \\ \frac{\rho}{\tan\theta}\frac{Y}{S^2} & -\frac{\rho}{\tan\theta}\frac{X}{S^2} & 1 \\ -\frac{Y}{S^2} & \frac{X}{S^2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.31}$$

and

$$\mathbf{E} = \begin{bmatrix} -\frac{h}{S}\rho R & \frac{\rho}{\tan\theta}\left(1-\frac{h}{S}Q\right) & -\frac{\rho}{q/p}\left(1-\frac{h}{S}Q\right) \\ \frac{\rho}{\tan\theta}\left(1-\frac{\rho}{S^2}Q\right) & \rho\left(\Delta\phi+\frac{\rho R}{S^2\tan^2\theta}\right) & \frac{\rho}{q/p\tan\theta}\left(\Delta\phi-\frac{\rho R}{S^2}\right) \\ \frac{\rho}{S^2}Q & -\frac{\rho}{S^2\tan\theta}R & \frac{\rho}{S^2 q/p}R \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.32}$$

where the abbreviations

$$\begin{aligned} X &= x_V - x_R - \rho\cos\left(\phi_V + \frac{\pi}{2}\right) \\ Y &= y_V - y_R - \rho\sin\left(\phi_V + \frac{\pi}{2}\right) \\ S &= \sqrt{X^2+Y^2} \end{aligned} \tag{5.33}$$

as well as

$$\begin{aligned} R &= Y\sin(\phi_V) + X\cos(\phi_V) \\ Q &= X\sin(\phi_V) - Y\cos(\phi_V) \end{aligned} \tag{5.34}$$

have been introduced.
The absolute term $\vec{c}_0$ in Eq. (5.25) can be calculated as

$$\vec{c}_0 = \vec{q}_0 - \mathbf{D}\,\vec{r}_0 - \mathbf{E}\,\vec{p}_0, \tag{5.35}$$

where $\vec{q}_0$ are the track perigee parameters with momentum $\vec{p}_0$ in point $\vec{r}_0$, which is chosen to be the point of closest approach along the helix with respect to the current best vertex estimate position R.
The equations of motion in the above presented approach are based on the assumption that the particle moves in an ideal constant magnetic field, which only points along the z-axis and therefore describes a perfect helical trajectory. While this is approximately true, and therefore sufficient, on short scales within the beam pipe for primary vertex reconstruction, this method is not very robust in all detector regions, where non-constant perturbations of the magnetic field and material effects can introduce larger uncertainties in the context of e.g. secondary vertex reconstruction.
A robust generalization of track parameter linearization, which is able to overcome these limitations while additionally incorporating timing information for time-dependent vertex fitting, has been developed in the course of this thesis and is discussed below.

5.3.5 Generalized Track Parameter Linearization

The above discussed helical track parameter linearization assumes an underlying track model with an ideal helical particle trajectory. As this assumption does generally not hold, especially for secondary vertexing in the presence of a potentially inhomogeneous magnetic field as well as material the particle can interact with, a new concept for a simple and robust semi-numerical generalized linearization approach, that additionally allows to easily incorporate a temporal component in the vertex fitting process, has been developed in the context of this thesis and is presented in the following.
Similar to the above presented linearized measurement equation (see Eq. (5.25)), the time-augmented linearized measurement equation can be expressed as

$$\vec{q}(\vec{r},\vec{p}) = \mathbf{D}\vec{r} + \mathbf{E}\vec{p} + \vec{c}_0, \tag{5.36}$$

with the time-dependent perigee parameters $\vec{q} = (d_0, z_0, \phi, \theta, q/p, t)^{\mathrm{T}}$, the space-time coordinates $\vec{r} = (x, y, z, t)^{\mathrm{T}}$ as well as the two Jacobians

$$\mathbf{D} = \frac{\partial(d_0, z_0, \phi, \theta, q/p, t)}{\partial(x, y, z, t)} \tag{5.37}$$

and

$$\mathbf{E} = \frac{\partial(d_0, z_0, \phi, \theta, q/p, t)}{\partial(\phi, \theta, q/p)}, \tag{5.38}$$

while the momentum vector is given by $\vec{p} = (\phi, \theta, q/p)^{\mathrm{T}}$.
Introducing a transformation in momentum representation from $\vec{p} = (\phi, \theta, q/p)^{\mathrm{T}} \longrightarrow \vec{p} = (T_x, T_y, T_z, q/p)^{\mathrm{T}}$ with the components

$$\begin{aligned} T_x &= \sin\theta\cos\phi \\ T_y &= \sin\theta\sin\phi \\ T_z &= \cos\theta \end{aligned} \tag{5.39}$$

of the momentum unit vector in spherical coordinates, Eq. (5.36) can be rewritten as

$$\vec{q}(\vec{r},\vec{p}) = \mathbf{D}\vec{r} + \underbrace{\begin{bmatrix} \frac{\partial d_0}{\partial T_x} & \frac{\partial d_0}{\partial T_y} & \frac{\partial d_0}{\partial T_z} & \frac{\partial d_0}{\partial (q/p)} \\ \frac{\partial z_0}{\partial T_x} & \frac{\partial z_0}{\partial T_y} & \frac{\partial z_0}{\partial T_z} & \frac{\partial z_0}{\partial (q/p)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial t}{\partial T_x} & \frac{\partial t}{\partial T_y} & \frac{\partial t}{\partial T_z} & \frac{\partial t}{\partial (q/p)} \end{bmatrix}}_{\mathbf{E}'} \cdot \underbrace{\begin{bmatrix} \frac{\partial T_x}{\partial \phi} & \frac{\partial T_x}{\partial \theta} & \frac{\partial T_x}{\partial (q/p)} \\ \frac{\partial T_y}{\partial \phi} & \frac{\partial T_y}{\partial \theta} & \frac{\partial T_y}{\partial (q/p)} \\ \frac{\partial T_z}{\partial \phi} & \frac{\partial T_z}{\partial \theta} & \frac{\partial T_z}{\partial (q/p)} \\ \frac{\partial (q/p)}{\partial \phi} & \frac{\partial (q/p)}{\partial \theta} & \frac{\partial (q/p)}{\partial (q/p)} \end{bmatrix}}_{\mathbf{F}} \cdot \vec{p} + \vec{c}_0 \tag{5.40}$$

by virtue of the chain rule for derivatives.
The transformation matrix $\mathbf{F}$ can easily be derived using Eq. (5.39) as

$$\mathbf{F} = \begin{bmatrix} -\sin\theta\sin\phi & \cos\theta\cos\phi & 0 \\ \sin\theta\cos\phi & \cos\theta\sin\phi & 0 \\ 0 & -\sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.41}$$

The combination of the matrices $\mathbf{D}$ and $\mathbf{E}'$

$$\begin{bmatrix} \mathbf{D} & \mathbf{E}' \end{bmatrix} = \begin{bmatrix} \frac{\partial d_0}{\partial x} & \cdots & \frac{\partial d_0}{\partial t} & \frac{\partial d_0}{\partial T_x} & \cdots & \frac{\partial d_0}{\partial (q/p)} \\ \frac{\partial z_0}{\partial x} & \cdots & \frac{\partial z_0}{\partial t} & \frac{\partial z_0}{\partial T_x} & \cdots & \frac{\partial z_0}{\partial (q/p)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial t}{\partial x} & \cdots & \frac{\partial t}{\partial t} & \frac{\partial t}{\partial T_x} & \cdots & \frac{\partial t}{\partial (q/p)} \end{bmatrix} = \frac{\partial \vec{q}_f^{\,\text{perigee}}}{\partial \vec{q}_i^{\,\text{free}}} = \mathbf{J}_{i\to f} \tag{5.42}$$

now exactly corresponds to the transport Jacobian $\mathbf{J}_{i\to f}$ from some initial parameter state i in free parameter representation $\vec{q}_i^{\,\text{free}} = (x, y, z, t, T_x, T_y, T_z, q/p)^{\mathrm{T}}$ to a final state f in perigee parametrization $\vec{q}_f^{\,\text{perigee}} = (d_0, z_0, \phi, \theta, q/p, t)^{\mathrm{T}}$.
As discussed in Section 4.2.3, extrapolation engines like the ATLAS extrapolator or the ACTS propagator (see Section 6.1.3) numerically derive these Jacobians for the transport of covariance matrices based on Runge-Kutta-Nyström integration. Thus, the desired Jacobians for track linearization $\mathbf{D}$ and $\mathbf{E} = \mathbf{E}' \cdot \mathbf{F}$ can easily be determined by exploiting an extrapolation engine to semi-numerically obtain $\mathbf{J}_{i\to f}$ without the need of analytically deriving the Jacobians based on an underlying track model.
The extrapolation of track parameters and their associated covariance matrices in an extrapolation engine typically involves the transport from one detector surface to another and the corresponding Jacobian $\mathbf{J}_{i\to f}$ therefore usually represents the propagation along a non-negligible path length with several consecutive Runge-Kutta-Nyström integration steps. However, the Jacobian $\mathbf{J}_{i\to f}$ desired for track linearization in the context of vertex reconstruction should ideally be evaluated in the close proximity to the linearization point of the track parameters. This is illustrated in Fig. 5.7, where (a) shows how track parameters in the perigee frame are first propagated outwards onto an arbitrary surface (here chosen to be the beam pipe surface) on which the track can be represented in free parametrization. The back-propagation to the perigee frame yields the desired Jacobian $\mathbf{J}_{i\to f} = \frac{\partial \vec{q}_f^{\,\text{perigee}}}{\partial \vec{q}_i^{\,\text{free}}}$. As the non-negligible spatial extrapolation from the beam pipe surface to the perigee is accounted for in $\mathbf{J}_{i\to f}$, the resulting linearization Jacobians $\mathbf{D}$ and $\mathbf{E}$ are not yet suitable for a linearization in the perigee point.
Only if the initial and final extrapolation positions are brought closer together, as illustrated in Fig. 5.7(b), i.e. the propagation path length s becomes very small and the extrapolation effectively equates to one single Runge-Kutta-Nyström integration step of the order of $\mathcal{O}(\mu\text{m})$, the resulting Jacobian $\mathbf{J}_{i\to f}$ accurately represents the dependency of the perigee parameters on the global position and momentum in the vicinity of the linearization point, as needed for the desired linearization.

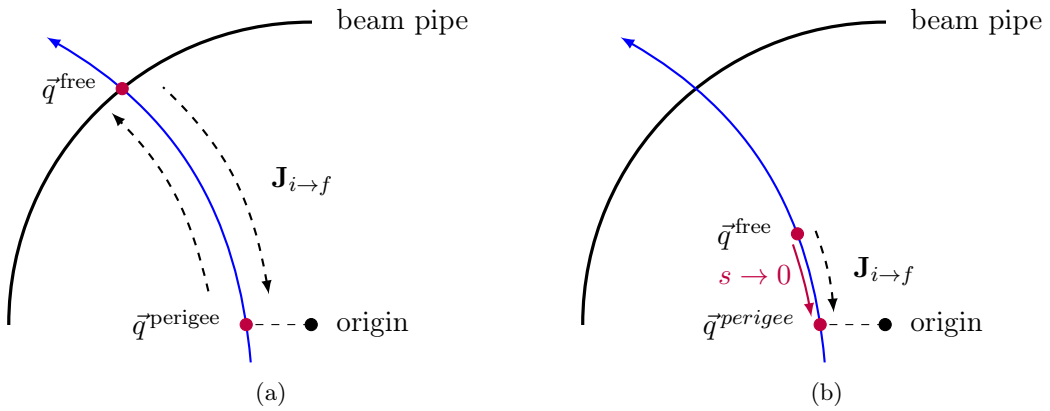


Figure 5.7: (a) Schematic illustration of the outwards propagation of perigee track parameters onto the beam pipe surface with a free track parametrization as well as the back-propagation into the perigee frame, where the Jacobian $\mathbf{J}_{i\to f}$ is derived. (b) Schematic illustration of the same propagation steps as depicted in (a), while the initial and final extrapolation positions are brought closer together with a propagation path length $s \to 0$ in order to retrieve the desired Jacobian $\mathbf{J}_{i\to f}$ in the vicinity of the linearization perigee point.

This newly presented approach does therefore not depend on the assumption of any underlying track model and purely estimates the linearized track parameters on a semi-numerical basis, allowing a robust linearization in all environments while natively incorporating timing information.
At the time of writing, the full functionality for the extrapolation with only one single Runge-Kutta-Nyström integration step is currently being developed in the ACTS propagator engine and will be fully available for allowing a robust and generalized time-dependent linearization of track parameters as described above in the near future. The adaptation of the Billoir and Kalman vertex fitting formalism to allow the full exploitation of time-dependent linearized track parameters for time-dependent vertex fitting, as e.g. required for the HGTD project as discussed in Section 3.3.2, is straightforward and has already been integrated in the newly developed vertexing software later presented in Chapter 6.

5.4 Primary Vertex Seed Finding

Vertex finding and vertex fitting are the two most important building blocks in almost any primary vertex finder algorithm. While the fitting part is performed by the vertex fitter, the actual vertex finding step, i.e. the first estimation of a vertex position, which can subsequently be fitted to determine its exact position, is done by a vertex *seed finder*. An effective and high-quality localization of these initial vertex estimates, called *vertex seeds*, is essential for the overall performance of most vertex reconstruction algorithms as vertices will be lost forever if no corresponding vertex seed can be found. Efficient and reliable vertex seed finder algorithms are therefore essential for the reconstruction of interaction vertices.

Four different vertex seed finding approaches for primary vertex reconstruction are discussed in the following. While the first two presented algorithms, the so-called Z-Scan Vertex Seed Finder and Gaussian Seeder, have been developed by ATLAS for primary vertexing in Run 1 to Run 3, two novel and highly efficient methods, the so-called Grid Seeder and Adaptive Grid Seeder, have been developed in the context of this thesis to overcome limitations seen in the previous approaches and are discussed in full detail as well.

5.4.1 Z-Scan Vertex Seed Finder

The *Z-Scan Vertex Seed Finder* [106] estimates a vertex seed position based on the z_0-impact parameter distribution of the seed track collection. While the seed vertex x- and y-coordinates are given by the beam spot center position in the transverse plane, the z-component is defined as the approximated mode value of all z_0 track parameters, estimated using a *Half Sample Mode* (HSM) algorithm [109].
Fig. 5.8 schematically illustrates how the HSM algorithm estimates a mode value of a given distribution with N entries without relying on a specific binning and therefore being entirely independent on the scale of the problem to be solved. The iterative procedure begins with finding all intervals along the z-axis that contain n data points (i.e. z_0-impact parameter values), where n is the smallest integer value that is greater or equal to $N/2$. The interval with the smallest length, and therefore highest data point density, is selected for the next iteration step, in which N is now set to the number of data points in the newly considered region, i.e. $N = n$. The procedure continues iteratively until only two data points are left and returns the arithmetic mean as the estimated mode value of the initial distribution.

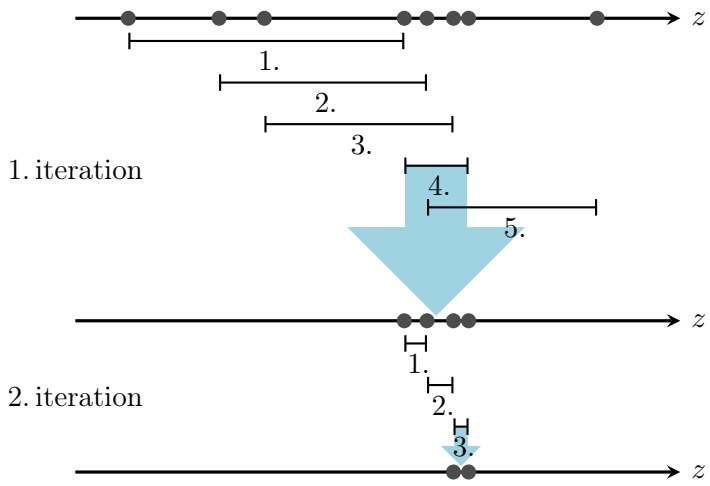


Figure 5.8: Schematic illustration of the half sample mode algorithm. Two iterations are shown in which the arrows depict the chosen down-sampling intervals.

The above described approach is therefore based on the assumption that a vertex candidate must be located in a region of high track density that is solely determined by the z_0-impact parameter values. In order to diminish the influence of secondary tracks, additional

information about the tracks' d_0-impact parameter values, and hence their compatibility with the beam spot, can be taken into account by weighting the contribution of each track to the z_0-distribution in the HSM algorithm. The Z-Scan Vertex Seed Finder iteratively finds intervals along the z-axis that contain at least 50% of the total weight, rather than 50% of the number of unweighted z_0 data points.
While this algorithm offers a simple and computationally inexpensive approach to vertex seed finding, it lacks the ability to incorporate information about individual track-related measurement errors. The impact parameters of all tracks contribute exactly the same amount to the mode finding procedure, irrespective of whether the track's parameters were measured very precisely or exhibit large uncertainties.
An approach to vertex seed finding that is able to overcome these limitation and therefore shows superb physics performance was developed by ATLAS and is discussed in the next section.

5.4.2 Gaussian Vertex Seed Finder

The *Gaussian Vertex Seed Finder* [103] (or short: *Gaussian Seeder*) models tracks as two-dimensional Gaussian distributions in the (d_0, z_0)-impact parameter plane to determine areas of high track density along the beam axis that can be interpreted as vertex candidate positions.
The track density of a track i at any position (r, z) in the (d_0, z_0)-plane is given by a radially and longitudinally correlated Gaussian probability distribution, centered around its respective impact parameter position $(d_{0,i}, z_{0,i})$ as

$$P_i(r,z) = \frac{1}{2\pi\sqrt{|\Sigma_i|}} e^{-\frac{1}{2}((r-d_{0,i}),(z-z_{0,i}))^T \Sigma_i^{-1} ((r-d_{0,i}),(z-z_{0,i}))}, \tag{5.43}$$

where the shape of the density distribution is determined by the track's covariance matrix

$$\Sigma_i = \begin{pmatrix} \sigma^2(d_{0,i}) & \sigma(d_{0,i}, z_{0,i}) \\ \sigma(d_{0,i}, z_{0,i}) & \sigma^2(z_{0,i}) \end{pmatrix}. \tag{5.44}$$

The total track density $W(z)$ at a given position z along the beam axis is calculated as the sum of all considered tracks

$$W(z) = \sum_{i \in \text{Tracks}} P_i(0, z). \tag{5.45}$$

Since Eq. (5.45) represent a sum of analytically known functions, the first and second derivative of $W(z)$ can be calculated at any z to find the global maximum of $W(z)$ and hence the position of highest track density along the beam axis, which will be interpreted as the seed vertex position, can be determined.
The procedure to find the global density maximum is based on the assumption that the vertex seed position must be in the vicinity of one of the given seed tracks and can be summarized as follows.

As illustrated in Algorithm 1, $W(z)$ and its first two derivatives $W'(z)$ and $W''(z)$ are calculated for every track at its respective z_0 value. If the curvature at the current trial position $z_{\text{trial}} = z_0$ is negative, i.e. $W''(z_{\text{trial}}) < 0$, and therefore z_{trial} is close to a maximum, an update step Δz towards the point with $W'(z) = 0$ is estimated and performed:

$$z_{\text{trial}} \to z_{\text{trial}} + \Delta z \tag{5.46}$$

As described in Ref. [103], the update step can either be performed as a simple Newton step

$$\Delta z = \frac{-W'(z)}{W''(z)} \tag{5.47}$$

or, if the underlying density function is assumed to be locally Gaussian-shaped, as

$$\Delta z = \frac{W(z)W'(z)}{W'^2(z) - W''(z)W(z)}. \tag{5.48}$$

In ATLAS, the latter step method is used as it shows better convergence over the Newton step approach.
The z position of the global maximum z_{max} is updated if $W(z_{\text{trial}})$ exceeds the previously highest density value (not explicitly shown in Algorithm 1 for simplicity). The above described process is repeated with up to two update steps in total and $W(z)$ as well as $W'(z)$ and $W''(z)$ are recalculated at the updated z_{trial} positions.
The $z = z_{\text{max}}$ position with the highest track density $W(z_{\text{max}})$ and $W''(z_{\text{max}}) < 0$ among all tracks is now considered the vertex seed z-position of interest.

Seed Width Estimation

An important property of the Gaussian Seeder is that it additionally allows to estimate the width σ_z of the vertex seed, which can be used to constrain the subsequent vertex fit and hence prevent unwanted influence of tracks from nearby vertices that might be compatible with the current vertex seed. Assuming a locally Gaussian track density distribution around the global maximum position z_{max}, the width can be expressed as

$$\sigma_z = \sqrt{-\frac{W(z_{\text{max}})}{W''(z_{\text{max}})}}. \tag{5.49}$$

Limitations

While the Gaussian Seeder provides state-of-the-art vertex seed finding features, such as the incorporation of individual track measurement uncertainties and the ability to estimate the seed's width, it comes along with the considerable limitation of not being well suited for high track-multiplicity events, as discussed in the following.
Every time a new vertex seed is needed by the vertex finder, the Gaussian Seeder performs a loop over all seed tracks in order to evaluate the maximum of the current track density

Set $z_{\text{max}} = 0$
foreach *track* i **do**
  Set $z_{\text{trial}} = z_{0,i}$
  Calculate $W(z_{\text{trial}})$, $W'(z_{\text{trial}})$ and $W''(z_{\text{trial}})$
  if $W''(z_{trial}) < 0$ **then**
    Potentially update maximum: $z_{\text{max}} = z_{\text{trial}}$
    Perform update step: $z_{\text{trial}} \rightarrow z_{\text{trial}} + \Delta z$
    Calculate $W(z_{\text{trial}})$, $W'(z_{\text{trial}})$ and $W''(z_{\text{trial}})$
  end
  if $W''(z_{trial}) < 0$ **then**
    Potentially update maximum: $z_{\text{max}} = z_{\text{trial}}$
    Perform update step: $z_{\text{trial}} \rightarrow z_{\text{trial}} + \Delta z$
    Calculate $W(z_{\text{trial}})$, $W'(z_{\text{trial}})$ and $W''(z_{\text{trial}})$
  end
  if $W''(z_{trial}) < 0$ **then**
    Potentially update maximum: $z_{\text{max}} = z_{\text{trial}}$
  end
end
Output: z_{max}

Algorithm 1: Simplified algorithmic illustration of how the Gaussian Seeder estimates the z-position of highest track density with two Newton-like update steps. The z_{max} position of the global maximum will only be updated if the current local maximum exceeds the previous global maximum value. Note that every calculation of $W(z_{\text{trial}})$ and its derivatives involves a full nested loop over all tracks.

distribution (see Algorithm 1). For every single track, $W(z)$ and its derivatives are calculated up to three times, where each calculation in turn requires a full nested loop over all considered tracks (see Eq. (5.45)).

The number of required track density calculations $P_i(0, z)$ (each involving a relatively expensive computation of an exponential function, see Eq. (5.43)) per vertex seed finder invocation with N seed tracks therefore scales with $\mathcal{O}(N^2)$, while the number of needed seed finder calls additionally increases with the presence of more vertices, i.e. higher pileup values and therefore generally higher track multiplicities. Although Ref. [103] claims that track-related information caching can be done in this approach, this only holds true up to a certain degree and the computationally most expensive part of this algorithm, i.e. recalculating $W(z_{\text{trial}})$ and its derivatives, becomes an execution path with many millions of invocations per event (also called *hot path*). The algorithm cannot make use of any cached information for $W(z_{\text{trial}})$ and its derivatives as the query z-positions always change slightly with every update step $z_{\text{trial}} \rightarrow z_{\text{trial}} + \Delta z$.

Fig. 5.9 shows the above mentioned scaling of the number of required track density calculations per event for different track multiplicities in $\langle\mu\rangle = 40 - 60$ $\mathrm{t\bar{t}}$ events. Given that these significant performance limitations are already seen for high track multiplicities (> 1000 tracks) in $\langle\mu\rangle = 40 - 60$ events, it is obvious that an optimized or novel vertex seed finding strategy is inevitable for ensuring acceptable reconstruction CPU times in the expected challenging HL-LHC environments with $\langle\mu\rangle \simeq 140 - 200$.

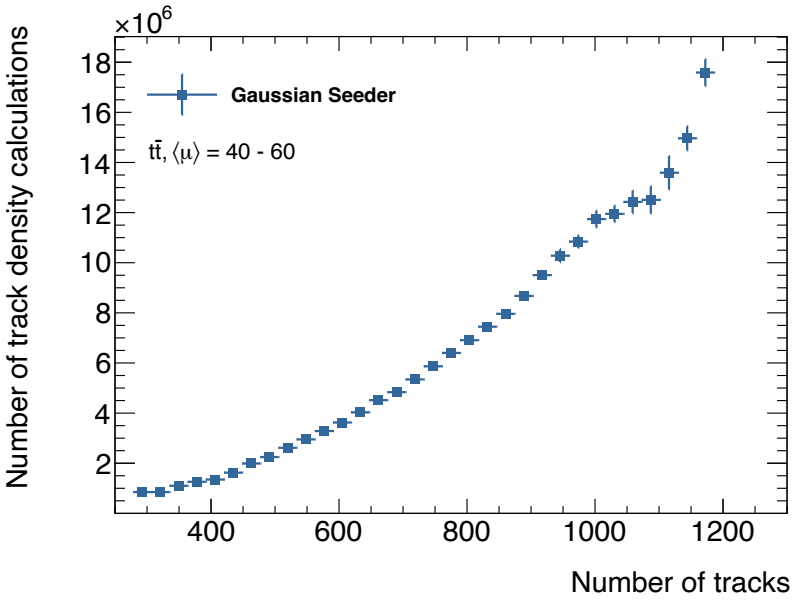


Figure 5.9: Number of track density calculations performed by the Gaussian Vertex Seed Finder per event for different track multiplicities in $\langle\mu\rangle = 40-60$ $t\bar{t}$ events. As the number of required track density calculations is dependent on both the track multiplicity as well as the number of seed finder invocations (i.e. the number of vertices per event), the small drop for $N_{\text{trk}} \gtrsim 1100$ tracks per event can be understood an artifact caused by the lack of events with more than ~ 60 vertices, given the upper bound of $\langle\mu\rangle = 60$ in the simulated MC samples. If events with $\langle\mu\rangle > 60$ and $N_{\text{trk}} \gtrsim 1100$ were present, the drop would disappear and the behavior observed for $N_{\text{trk}} \lesssim 1100$ would continue also in this region.

5.4.3 Grid Vertex Seed Finder

The Gaussian Seed Finder introduced in Section 5.4.2 provides state-of-the-art vertex seed finding features that will be needed to ensure excellent physics performances in the upcoming high vertex-density scenarios at the HL-LHC and other future particle accelerators. However, its severe limitation of not being able to make use of cached track information results in drastically increasing CPU consumptions in these dense environments, excluding it as a possible candidate for vertex seed finding in the challenging conditions after LHC Run 3.

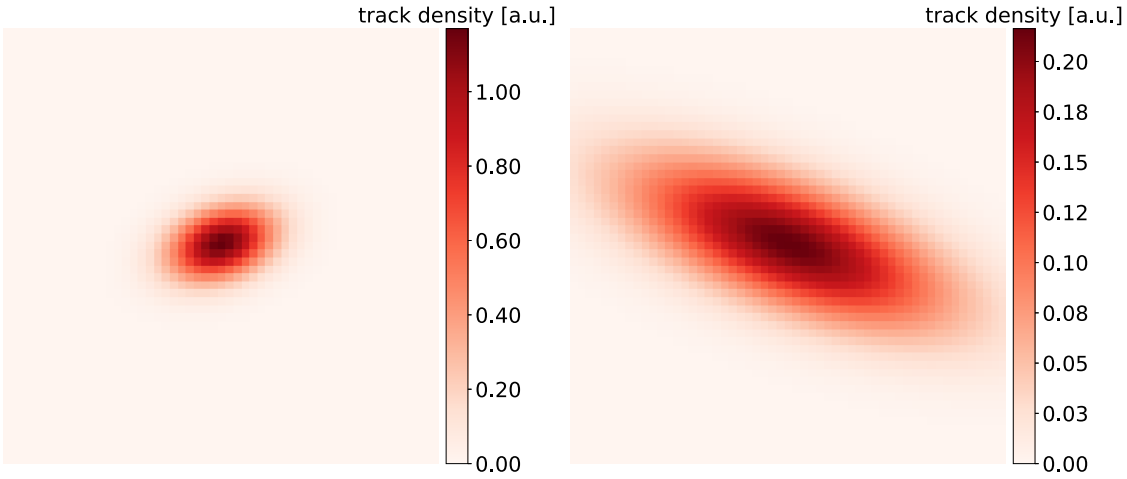
In order to maintain the excellent physics performance of the Gaussian Seeder, while additionally ensuring superb CPU performances in high pile-up environments, a novel approach to vertex seed finding, the so-called *Grid Vertex Seed Finder* (or short: *Grid Seeder*), was developed in the context of this thesis and will be presented in the following.

Basic Concept of Track Density Grids

Similar to the Gaussian Seeder, the Grid Seeder models tracks as two-dimensional Gaussian distributions in the (d_0, z_0)-impact parameter plane, centered around their respective track impact parameter points (d_0, z_0), where the width and shape of the distributions are determined by the track covariance matrices.

In this newly developed approach, however, the (d_0, z_0)-impact parameter plane is subdivided into a two-dimensional grid with configurable binning of size w, making the track representations themselves also two-dimensional Gaussian track density grids. Example

grid representations of two single tracks with different shapes and sizes in the (d_0, z_0)-impact parameter plane are shown in Fig. 5.10. The size of these squared track representation grids is given by the number of bins in d_0 and z_0 direction, called $N_{\text{bins}}^{\text{trk}}$, and is also a configurable parameter of the algorithm.



(a) Grid density distribution of a single track with $\sigma^2(d_0) = 0.1\,\text{mm}^2$, $\sigma^2(z_0) = 0.2\,\text{mm}^2$ and $\sigma(d_0, z_0) = 0.02\,\text{mm}^2$.

(b) Grid density distribution of a single track with $\sigma^2(d_0) = 0.5\,\text{mm}^2$, $\sigma^2(z_0) = 1.8\,\text{mm}^2$ and $\sigma(d_0, z_0) = -0.1\,\text{mm}^2$.

Figure 5.10: Examples of two track density representations on a grid in the (d_0, z_0)-impact parameter plane with a grid size of $N_{\text{bins}}^{\text{trk}} \times N_{\text{bins}}^{\text{trk}} = 55 \times 55$ bins and bin width of $w = 0.1\,\text{mm}$.

Comparing Fig. 5.10(a) and Fig. 5.10(b) with their respective $\sigma^2(d_0)$ and $\sigma^2(z_0)$ values, is it apparent that larger d_0 and z_0 uncertainty values in the track covariance matrix lead to broader density distributions in the (d_0, z_0)-impact parameter plane, whereas the rotation is determined by $\sigma(d_0, z_0)$.
The exact choice of the $N_{\text{bins}}^{\text{trk}}$ and w parameters to fully contain the vast majority of track density distributions within the grid representation, and therefore guarantee highest possible physics performances, is of course highly dependent on its application and will be further discussed in the context of its ACTS and ATLAS integration in Chapter 7 as well as in Appendix A.6.

Construction of the Event Track Density Distribution

In the process of primary vertex finding only vertices along the beam axis are of interest. Therefore, the Grid Seeder does *not* need to evaluate all density values in the entire two-dimensional density grid for each track as shown above in Fig. 5.10, but instead only calculates track density contributions along the beam axis:
Any given track is placed in the (d_0, z_0)-plane, centered around its respective impact parameter point (d_0, z_0), and the track density values in the overlap vector of its density representation grid with the central beam axis vector, i.e. the contribution of a single track to the overall track density distribution along the beam axis, can be determined as illustrated in Fig. 5.11.
The contribution of a single track to the overall track density is therefore fully specified

by a vector of length $N_{\text{bins}}^{\text{trk}}$ (depicted in red), as well as the z-position at which the density contribution vector was placed along the z-axis. The Grid Seeder therefore reduces the problem of vertex seed finding with Gaussian track densities to be purely one-dimensional with overlap density vectors as track representations. The track density vectors of all tracks in the seed track collection can now be superimposed along the beam axis and the bin with the highest track density, usually the one considered to be in the position of a vertex estimate, can be easily determined.

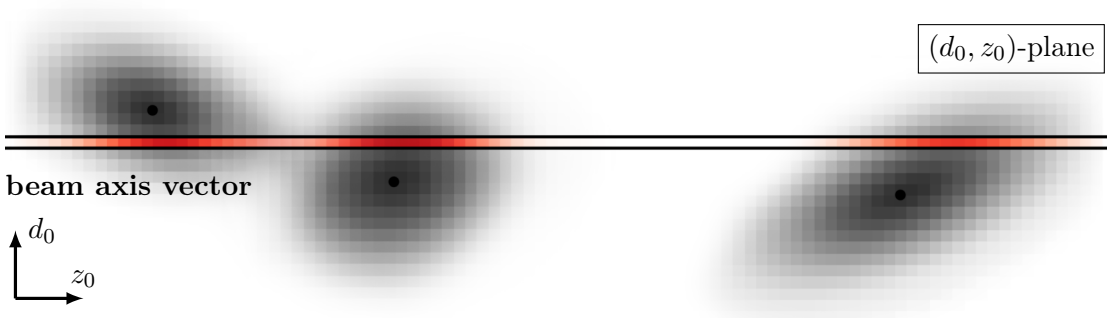


Figure 5.11: Three single tracks in the d_0-z_0-plane, represented as two-dimensional Gaussian density grids, centered around their respective impact parameter points (black dot). Only the density values in the overlap vectors between the track grids and the z-axis are calculated and superimposed (red). Published in Ref. [110].

The beam axis density vector containing all superimposed track densities is given by a fixed-size vector $\mathcal{D} \in \mathbb{R}^{N_{\text{bins}}}$ with N_{bins} bins. It covers a z-region in the interval $[-z_{\text{max}}, z_{\text{max}}]$, where the absolute value of the maximum z-coverage z_{max} is determined by the chosen bin width w and the number of bins along the beam axis N_{bins} according to

$$z_{\text{max}} = \frac{N_{\text{bins}}}{2} \times w. \tag{5.50}$$

An adequate choice of these parameters is of course highly experiment-specific, depending on the z-region to be covered as well as the desired seed position resolution. A z-region coverage of for instance $z_{\text{max}} = 250\,\text{mm}$, which could be an appropriate choice if employed in ATLAS, with a bin-width of $w = 0.1\,\text{mm}$ would be achieved by utilizing a 5000-dimensional fixed-sized vector (i.e. $N_{\text{bins}} = 5000$) containing all superimposed track densities along the beam axis.

A more detailed study on the choice of these parameters in the context of primary vertex reconstruction in ATLAS will be presented in Chapter 7.

Having the fixed-sized vector $\mathcal{D}$ covering a certain z-region comes with the advantage that each bin, or rather its center, directly corresponds to a specific z-position and the vector $\mathcal{D}$ can therefore work as a ruler that immediately encodes all necessary z-position information. Thus, no further information other than the track density values in $\mathcal{D}$ need to be stored for vertex seed finding.

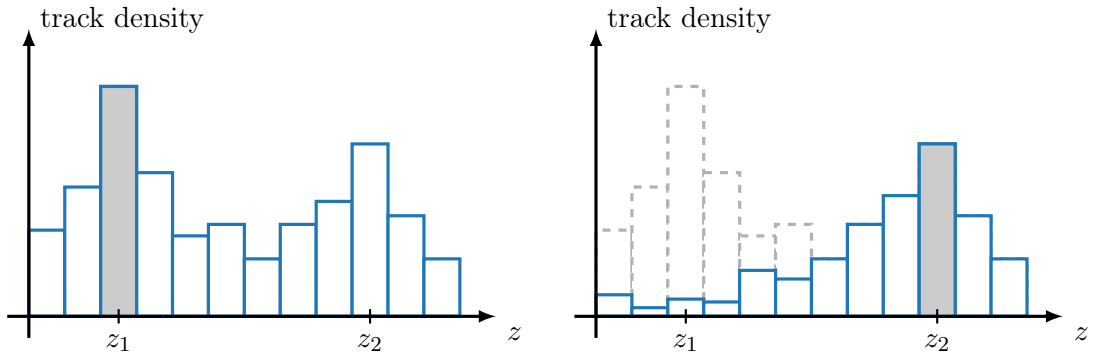
Vertex Seed Position Estimation

After superimposing the track density vectors of all seed tracks and hence constructing the overall track density distribution along the beam axis, the seed vertex z-position can easily be determined as the z-position of highest track density.
The most straightforward approach to finding the highest track density z-position along the beam axis is to just locate the maximum bin in $\mathcal{D}$ and define the z-value of its center as the seed vertex position of interest.

Information Caching and Track Removal

The approach of effectively representing every track as a simple overlap density vector with the beam axis can be used to take full advantage of track-related information caching, which immensely reduces the number of necessary computations and can therefore result in great CPU performance improvements, as explained in the following.
In the iterative process of primary vertex finding, the vertex finder algorithm typically first invokes the seed finder on a set of seed tracks to construct a first vertex estimate, which is then fitted using all available tracks that are compatible with the current position. In every subsequent iteration, the seed finder is again invoked to find a new seed with the exact same seed track collection input, only reduced by the tracks that have been used for the fit in the previous iteration. Instead of constructing the overall track density distribution along the beam axis from scratch every single iteration (which will be an almost exact copy of the one from the previous iteration, only reduced by the track density contribution of the removed compatible tracks), the overall distribution along the beam axis is only constructed once in the very first iteration and density contributions of tracks used in the fit are iteratively removed in all subsequent iterations:
In the very first iteration of the vertex finding process, where still all seed tracks are available for vertex seed finding (i.e. they are not yet associated to other vertices and therefore not yet removed from the seed track collection), the overall event track density distribution along the beam axis $\mathcal{D}_1$ can be calculated by superimposing all individual track densities as explained above. Fig. 5.12(a) shows an example of a resulting track density distribution along the beam axis. As every track can be represented as a vector of density values, it is extremely easy to store these individual representations and use them in subsequent iterations: After the first vertex seed is found and the vertex finder removed its associated tracks from the seed track collection, the track density contributions of the removed seed tracks can simply be deducted from $\mathcal{D}_1$ at their respective positions to obtain the overall track density distribution in the second iteration $\mathcal{D}_2$. Fig. 5.12(b) shows the example track density distribution in the subsequent iteration after the removal of the seed tracks associated to the vertex found in the first iteration.
Consequently, this newly presented approach to primary vertex seed finding does not need to rely on recalculating and adding up individual track densities in every single vertex finder iteration. Instead, it calculates the overall event track density distribution only once in the very first iteration and uses simple and easily cacheable track representations

to iteratively remove tracks from the density distribution $\mathcal{D}$ until no seed tracks are left, i.e. $\mathcal{D}$ is empty, and the vertex finding process is completed.



(a) Track density distribution $\mathcal{D}_1$ (b) Track density distribution $\mathcal{D}_2$

Figure 5.12: Example of an event track density distribution (a) before and (b) after the removal of seed tracks associated to the vertex seed at $z = z_1$. The dark grey bins represent the current maximums at the given iteration and the light grey dashed bins in (b) the removed track density contributions.

The number of required track density calculations, the computationally most expensive part of the entire vertex seed finding process, will therefore scale only linearly with the number of seed tracks, as for each single track $N_{\text{bins}}^{\text{trk}}$ density calculations need to be performed once per event. As shown in Fig. 5.13, this results in a significantly lower number of required track density calculations compared to the Gaussian Seeder discussed in Section 5.4.2. Especially for high track multiplicities, a reduction factor of more than two orders of magnitude can be seen, making the Grid Seeder an excellent choice for high pile-up environments. A full in-detail analysis of the Grid Seeder's CPU performance will be given in Chapter 7.

Seed Width Estimation

In addition to the seed vertex z-position, another useful and important quantity for accurate vertex finding and fitting is the width or uncertainty of the seed vertex, that can be used to better select tracks that are compatible with the current vertex estimate and to constrain the exact vertex position during the vertex fit. A seed width estimation technique in the context of the Grid Vertex Seed Finder has been developed and will be presented in the following.

Assuming a Gaussian-like track density distribution around the current maximum of interest, the distribution's standard deviation, and hence its width, σ_z can be determined based on the *full width at half maximum* (FWHM):

The probability density function of a one-dimensional normally distributed variable z with

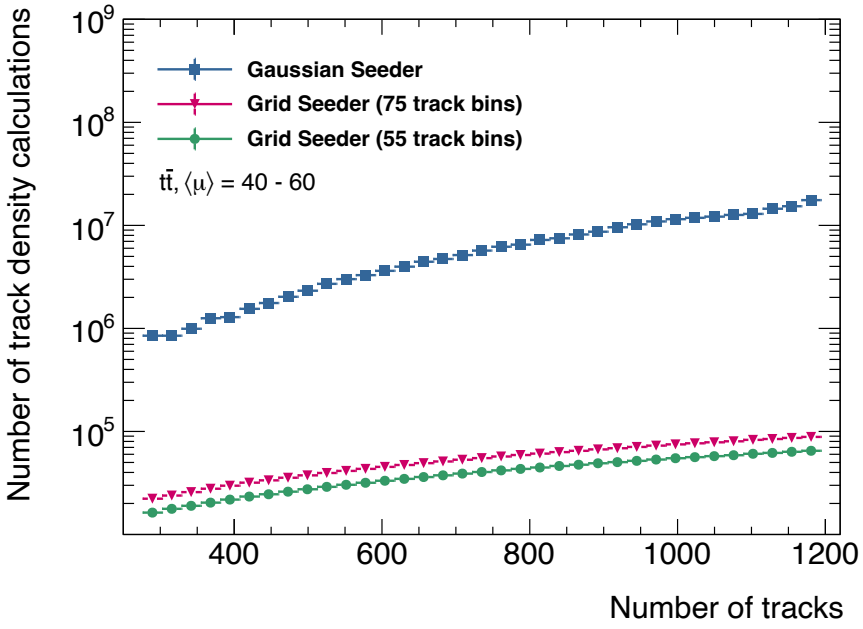


Figure 5.13: Number of track density calculations performed by the Gaussian Seed Finder and the Grid Seed Finder for $N_{\text{bins}}^{\text{trk}} = 75$ and $N_{\text{bins}}^{\text{trk}} = 55$ per event for different track multiplicities in $\langle\mu\rangle = 40 - 60$ $\text{t}\bar{\text{t}}$ events.

a mean value μ and a standard deviation σ_z is generally given by

$$f(z) = \frac{1}{\sqrt{2\pi\sigma_z^2}} e^{-\frac{(z-\mu)^2}{2\sigma_z^2}}. \tag{5.51}$$

The half-maximum points $z_\pm$ can be found by solving

$$f(z_\pm) = \frac{1}{2} f(\mu) \tag{5.52}$$

with the maximum of the distribution being located at $z_{\text{max}} = \mu$ with $f(z_{\text{max}}) = f(\mu)$. Inserting the definition of Eq. (5.51) into Eq. (5.52), we find

$$e^{-\frac{(z_\pm-\mu)^2}{2\sigma_z^2}} = \frac{1}{2} e^{-\frac{(\mu-\mu)^2}{2\sigma_z^2}} = \frac{1}{2} \tag{5.53}$$

and therefore

$$-\frac{(z_\pm - \mu)^2}{2\sigma_z^2} = -\ln 2, \tag{5.54}$$

resulting in the half-maximum points to be given by

$$z_\pm = \pm\sqrt{2\ln 2}\,\sigma_z + \mu. \tag{5.55}$$

The full width at half maximum can now be expressed as a function of σ_z

$$\text{FWHM}(\sigma_z) \equiv z_+ - z_- = \sqrt{8\ln 2}\,\sigma_z \approx 2.35\,\sigma_z. \tag{5.56}$$

Having the relationship in Eq. (5.56) established, the FWHM of the grid track density distribution can now be approximated and subsequently used to determine the vertex seed's width σ_z. Fig. 5.14 illustrates the FWHM approximation method developed for the Grid Seeder, which starts at the current full maximum of interest and proceeds in both directions until the first bins with density values below the half-maximum are found on both sides (depicted as grey bins).

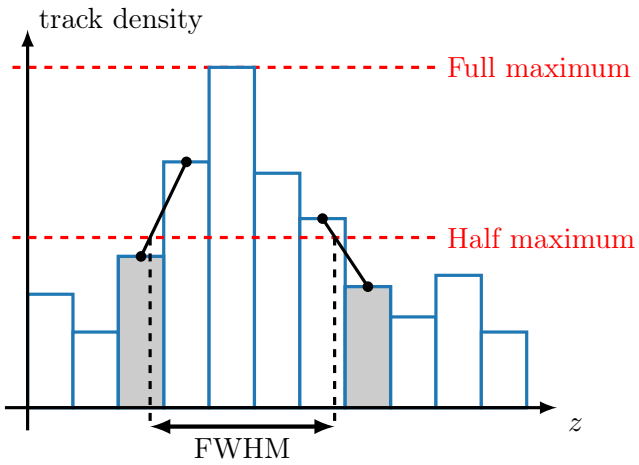


Figure 5.14: Illustration of the approximation method used to determine the FWHM of the grid track density distribution. A linear approximation (black line) between the bins right below the half maximum (grey bins) and the previous bins are used to find the intersection positions with the half-maximum line to eventually identify the FWHM.

A linear approximation of the underlying Gaussian distribution is now employed between the current bin below the half-maximum and the previous bin, right above the half maximum (illustrated as black lines). The intersection z-positions of these linear approximations with the half-maximum line set the lower and upper z-limit, z_- and z_+, for the FWHM determination.
The seed width σ_z can now be evaluated using the inverse of Eq. (5.56) and the approximated FWHM of the grid track density distribution:

$$\sigma_z = \frac{\text{FWHM}}{\sqrt{8 \ln 2}} \tag{5.57}$$

Memory Utilization

In addition to an excellent CPU performance, an efficient memory management is important when employing reconstruction tools in HEP experiments. The memory impact of the above presented Grid Seeder method will therefore be briefly discussed in the following.
The Grid Seeder is based on a fixed-size vector $\mathcal{D} \in \mathbb{R}^{N_{\text{bins}}}$ that stores all track density values along the beam axis and simultaneously functions as a ruler to directly encode their respective z-positions. While this approach is beneficial in terms of simplicity and computational speed as only one fixed-size block of memory needs to be allocated and can be filled over time, it can result in a relatively large amount of allocated bins, and therefore

memory, that remains unused throughout the entire seed finding process. The total number of required bins along the beam axis N_{bins} is determined by two factors: First, the maximum z-coverage along the beam axis z_{max} needs to be defined in such a way that all possible reconstructed tracks are always captured by the density grid. The Grid Seeder is only sensitive to the interval $[-z_{\text{max}}, z_{\text{max}}]$ and with these hard z-boundaries applied, everything outside of this area will fall out of acceptance. The exact choice of z_{max} is of course highly experiment-specific but in general a large value is desired in order to cover a broad z-range and not lose any tracks and therefore vertices.
Additionally, the z-position resolution of the algorithm is governed by the bin size w, for which very small values are needed in order to guarantee a good physics performance. According to Eq. (5.50), the dimension of the density vector $\mathcal{D}$, i.e. the number of required bins along the z-axis, needs to be high to ensure a good position resolution while covering a large z-region of interest.
At the same time, the beam spot in collider experiments usually follows a Gaussian distribution with only a very few primary interactions taking place in the tails of the distribution, generally leaving a large amount of available bins at large z-values unused and entirely empty.
Fig. 5.15 shows the fractions of non-empty density bins per event for different N_{bins} values and the corresponding bin widths w with a fixed z-coverage of $z_{\text{max}} = 250\,\text{mm}$, tested on ATLAS $\langle\mu\rangle = 40 - 60$ $\text{t}\bar{\text{t}}$ events. With increasing resolution (i.e. higher values of N_{bins}) the average fraction of filled density bins drop to about 20% for $N_{\text{bins}} = 20\,000$, meaning that a vast majority of the allocated fixed-size memory block remains unused on average.
In the actual implementation of this algorithm (see Chapter 6), each density bin is represented by a `float` number with a typical size of 4 bytes. Therefore, even a high-granularity configuration of the Grid Seeder with $N_{\text{bins}} = 20\,000$ would only require $20\,000 \times 4\,\text{bytes} = 80\,\text{kb}$ of allocated memory for the total beam axis density vector, a number that is almost negligible compared to the tens or hundreds of megabytes that are usually needed by reconstruction software. Having a large proportion of density bins allocated but not used will thus in general not be crucial in terms of memory efficiency.

However, in order not to restrict ourselves to specific use cases and also provide a state-of-the-start vertex seed finding method in case that extremely efficient memory management is important, a very large area needs to be covered with high granularity (i.e. $N_{\text{bins}} \gg 20\,000$ is needed) or a fixed z-coverage with a constant z_{max} is not practical, an adaptive version of the Grid Seeder, which dynamically grows its density vector $\mathcal{D}$ along the z-axis over time, has also been developed in the course of this thesis and will be presented in Section 5.4.4.

Summary of the Presented Approach

In summary, the above presented approach is able to provide state-of-the-art vertex seed finding features, such as the ability to estimate the seed's width and incorporate individual track measurement uncertainties in the finding process. At the same time, it overcomes

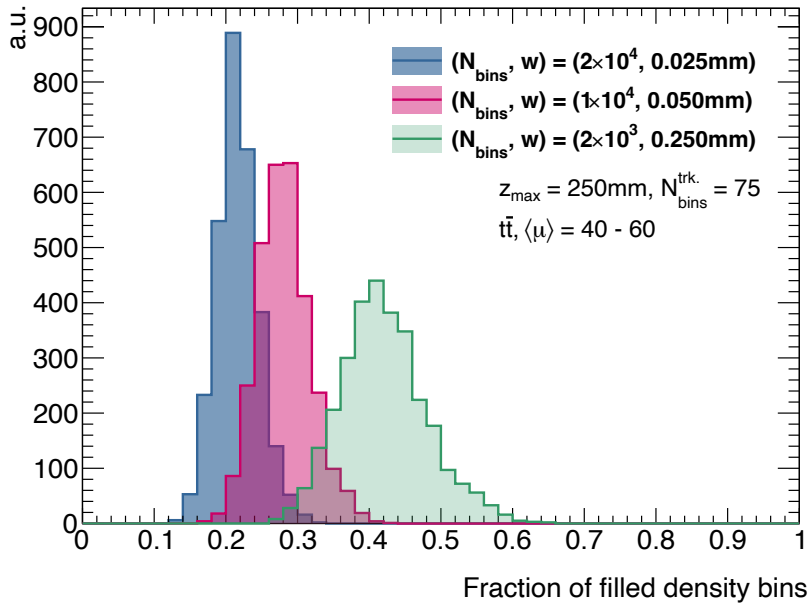


Figure 5.15: Fraction of filled density bins per event for different numbers of density bins along the z-axis in $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

the Gaussian Seeder's crucial limitation of not being able to cache computationally expensive track density information by making use of easily cacheable track representation vectors. Since the method is based on static, fixed-size memory allocation, superb CPU performances especially in high pile-up environments are expected, which come, however, with the potential trade-off of having to allocate memory that remains unused throughout the seed finding process and a required pre-defined z-coverage interval.

5.4.4 Adaptive Grid Vertex Seed Finder

This section presents an adaptive and thus more flexible version of the above discussed Grid Seeder algorithm, the *Adaptive Grid Vertex Seed Finder*, or short Adaptive Grid Seeder, which allows for a dynamic growth of the density vector $\mathcal{D}$ along the z-axis as more tracks are added to the system.

Construction of the Event Track Density Distribution

The concept of individual tracks being represented as track density vectors resulting from the overlap of a two-dimensional track density grid with the beam axis remains exactly the same as for the non-adaptive Grid Seeder algorithm. The fundamental difference lies in the way these track representations are used to construct the overall event track density distribution $\mathcal{D}$.

While the Grid Seeder places individual tracks at their respective positions on a pre-defined fixed-size vector covering the entire z-region of interest with hard boundaries, the Adaptive Grid Seeder starts from a zero-dimensional, i.e. empty, event track density vector $\mathcal{D}$, which

grows over time as new track densities are added, as explained in the following.
Before any tracks are added, the algorithm starts off with two zero-dimensional vectors, one for eventually storing the overall event track density distribution, the other for storing its corresponding z-position values. As shown in an exemplary illustration in Fig. 5.16(a), each newly added track is now given by an $N_{\text{bins}}^{\text{trk}}$-dimensional density vector representation (with $N_{\text{bins}}^{\text{trk}} = 5$ in the example) together with an additional same-size vector storing their corresponding z-positions. After the first track was added to the overall event track density vector, its size has grown to $N_{\text{bins}}^{\text{trk}}$, holding the exact same content as the newly added track. If a second track is subsequently added to the event track density vector, the z-position of its density bins are evaluated and inserted in ascending order at the respective z-positions in the event density vector (shown in Fig. 5.16(b)). The event density vector grows in size accordingly.
In case of overlapping density bins, i.e. a z-bin of the newly added track already exists in the event track density vector, the new components are just added to the already existing density bin. This is illustrated in Fig. 5.16(c), where the new track shares three of its five bins with the first track. While the overlapping density values are summed up, the two non-overlapping bins are again just inserted at their respective position, making the event track density vector grow by two entries.

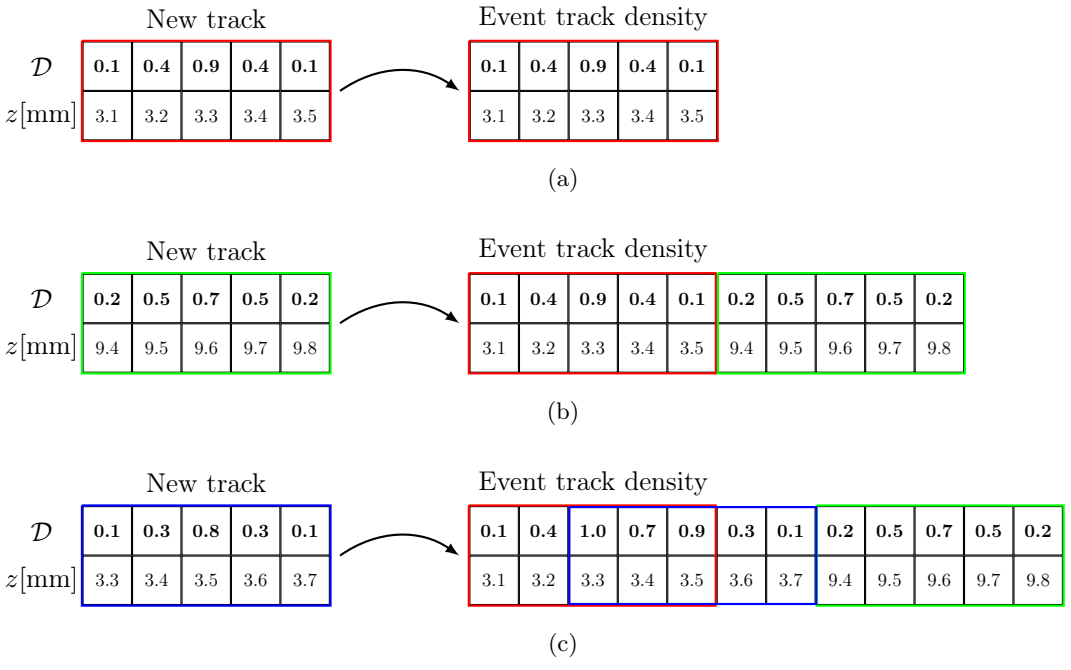


Figure 5.16: Exemplary illustration of the adaptive growth of the event track density vector in the Adaptive Grid Seeder. Non-overlapping track densities are inserted at their respective z-positions, see (a) and (b), while density components of newly added tracks at already existing z-bins are summed up (c).

Information Caching, Seed Vertex Position and Width Estimation

After the event track density distribution has been constructed, the maximum z-position as well as the seed width can be determined in the exact same manner as for the non-adaptive Grid Seeder. Note that the resulting non-zero event track density distribution will be identical to the one of the non-adaptive method. Position and width results, and thus the physics performance, will therefore be exactly identical for the non-adaptive and adaptive Grid Seeder algorithms.
Furthermore, all track-related information can also be cached and track densities can easily and iteratively be removed if the corresponding tracks have been removed from the seed track collection. While the event track density vector grows dynamically in size when tracks are added, its dimension stays the same for CPU efficiency reasons when tracks are again removed.
Since the z-positions are not directly encoded in a ruler-like vector along the beam axis any longer (as for the Grid Seeder), more information needs to be stored in an additional vector holding all corresponding z-values. While this leads to the allocation of twice as much memory per track density bin compared to the non-adaptive Grid Seeder, a certain amount of memory can nonetheless be saved, especially when high z-resolutions are desired and the resulting small bin sizes would otherwise lead to a large amount of unused density bins, as was shown for the Grid Seeder in Fig. 5.15.
The Adaptive Grid Seeder therefore serves as a flexible and slightly memory-optimized alternative to its non-adaptive counterpart without the need of pre-defining hard z-boundaries. As it is based on dynamic memory allocation with adaptively growing vectors, which is generally less CPU efficient compared to utilizing static memory as in the Grid Seeder case, a slight degradation in CPU performance is expected.
The choice between the application of the non-adaptive or adaptive version of the Grid Seeder will therefore be a trade-off between memory and CPU efficiency as well as the need of having flexible z-boundaries and will therefore strongly depend on its use case.
In any case, both of these newly presented primary vertex seed finders provide state-of-the-art features that, together with their elaborated algorithmic properties, will allow for outstanding physics as well as CPU performances.
A full in-depth analysis of the physics as well as CPU performance of the Grid Seeder and Adaptive Grid Seeder algorithm will also be presented in Chapter 7.

5.5 Reconstruction of Secondary Vertices

In addition to the precise reconstruction and understanding of the primary proton-proton interaction points as discussed above, the knowledge of displaced, secondary vertices arising from e.g. decays of long-lived hadrons is vital for many downstream algorithms, such as heavy-flavor tagging, and thus important for many physics analyses. Heavy-flavor tagging refers to the identification of particle jets containing b-flavored or c-flavored hadrons and is an important tool for many physics analyses, such as the one presented in this thesis in Chapter 9.

This section first briefly discusses one commonly used approach to secondary vertex reconstruction in ATLAS, the so-called *JetFitter* algorithm, followed by the application of secondary vertexing in the context of heavy-flavor tagging.

5.5.1 The JetFitter Secondary Vertexing Algorithm

Secondary vertex reconstruction using the JetFitter algorithm [111, 112] is based on several different reconstructed input objects, such as ID tracks after some dedicated quality cuts[1], primary vertex information (in particular the selected hard-scatter vertex) as well as Anti-k_t clustered hadronic jets with a radius parameter of $R = 0.4$.
The main goal of the JetFitter algorithm, as employed in ATLAS, is now to find and reconstruct secondary vertices within a reconstructed jet that can be identified as decay vertices of long-lived b-flavored hadrons. The typical decay topologies of b-flavored hadrons (B-hadrons) are given by their relatively long lifetimes ($< c\tau > \simeq 0.5\,\text{mm}$) and their subsequent decays, preferably into c-flavored hadrons (D-hadrons), which in turn undergo a weak decay after a non-negligible life time. The JetFitter algorithm makes use of this distinct topology and is thus based on the assumption that the primary hard-scatter vertex as well as the two secondary vertices to be reconstructed, the B-decay vertex and the D-decay vertex, all lie on the same line, given by the flight axis of the B-hadron. All charged particle tracks participating in the vertex fits are expected to have originated from the B- or D-decay and are therefore required to intersect with the B-hadron flight axis. The expected relation between the primary vertex, the B flight axis as well as the two secondary decay vertices is schematically shown in Fig. 5.17. The B-hadron flight axis is initialized by the jet axis of the reconstructed Anti-k_t jet under consideration and eventually subject to the vertex fit itself.
The vertex fit is implemented as a Kalman filter with an additional line constraint, allowing to perform a fit for every single considered track with the hypothesis that each intersection of a track and the B-hadron flight axis represents a single vertex. Once a number of fitted vertices along the flight axis has been found, in the first iteration purely stemming from intersections of single tracks with the axis, the most compatible pair of vertices can be merged and a vertex refit is performed. This procedure can be continued iteratively, until no mergeable pairs of vertices are left and, ideally, the B- and D-decay vertices have been found. This approach has the advantage that also incomplete topologies, where only e.g. a single reconstructed charged particle track associated to one of the two cascade vertices is available, can be reconstructed, making it a flexible and robust algorithm for the reconstruction of secondary vertices and their interpretation in the context of heavy-flavor tagging.

5.5.2 Application for Heavy-Flavor Tagging

In order to gain a better understanding of the underlying physics process and to therefore increase the discrimination power between heavy-flavor jets and so-called light jets (i.e. jets

[1] These cuts are typically very similar to the ones presented in Section 5.2.1 for primary vertexing, but are usually made orthogonal by inverting the selection criteria on the transverse impact parameter $|d_0|$.

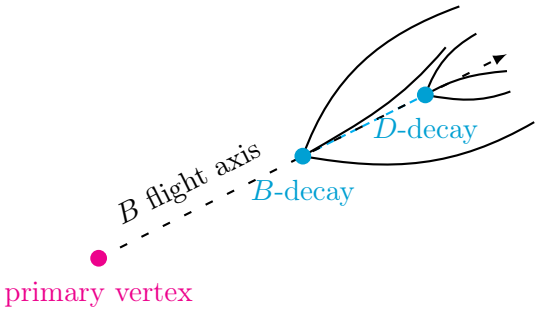


Figure 5.17: Schematic illustration of a typical B-hadron decay topology, which is assumed in the JetFitter algorithm. The primary hard-scatter vertex, the B- and resulting D-decay vertices are all aligned on the B-hadron flight axis.

originating from u-, d-, s-quarks), the cascade topologies reconstructed by the JetFitter algorithm can be further evaluated. Several discriminating variables, including the invariant masses of tracks associated with the decay vertices, the average three-dimensional decay length significance as well as the track multiplicities at the decay vertices, are determined and can be used as input to higher-level heavy-flavor classification algorithms [113].

In practice, ATLAS uses a combination of multiple different low-level reconstruction and classification algorithms, such as the `IP2D` and `IP3D` taggers [114], making use of the impact parameter significances to construct discriminating variables, the secondary vertex tagging algorithm `SV1` [115], reconstructing a single displaced vertex within a jet based on a χ^2-minimization, as well as the above discussed topological JetFitter algorithm. All discriminating variables constructed by these low-level algorithms are used as input to multivariate classifiers, which are trained to distinguish between b-jets, c-jets and light-flavor jets. The two commonly employed classification methods in ATLAS are the `MV2` algorithm [116], which is based on a dedicated boosted decision tree architecture, as well as the `DL1` algorithm [116], based on a deep feed-forward neural network.

A variation of the `DL1` algorithm, that additionally uses information from the recurrent neural network `RNNIP` algorithm [117] as input, the so-called `DL1r` algorithm, will be the b-tagging method of choice in the analysis presented in Chapter 9.

The multi-class classification neural network of the `DL1r` algorithm is trained to predict the probability of a jet to be a b-jet, c-jet or light-jet, labeled as p_b, p_c and p_u, respectively. These predicted jet-flavor probabilities are used to define the final `DL1r` b-tagging discriminant as

$$\texttt{DL1r} = \ln\left(\frac{p_b}{f_c \cdot p_c + (1 - f_c) \cdot p_u}\right), \tag{5.58}$$

where both the cut on `DL1r` as well as f_c are adjustable parameters of the algorithm that can be used to tune the desired b-tagging efficiency.

As further discussed in the context of the presented analysis in Chapter 9, this algorithm can also be adapted for the identification of c-jets by interchanging the b-jet and c-jet probabilities p_b and p_c in the definition of the `DL1r` quantity.

Chapter 6

Implementation of the ACTS Vertexing Software

A modern, highly performant and thread-safe primary vertex reconstruction software suite has been developed in the course of this thesis.
The software was implemented within the experiment-independent ACTS software toolkit, a toolkit of previously tracking-only software components, and provides a variety of production-ready vertex reconstruction algorithms which are, at the time of writing, already being used by several different HEP experiments. This chapter discusses various aspects of the implementation of the newly developed ACTS vertexing software suite.

6.1 ACTS – A Common Tracking Software

6.1.1 Introduction

The ACTS (A Common Tracking Software) [118] project, initiated in 2016 at CERN, aims to provide a modern, detector-independent software toolkit for track and vertex reconstruction. It is based on the well-tested reconstruction code base of the ATLAS experiment and is designed to be inherently thread-safe to support parallel code execution in a generic and framework-independent manner. All implemented algorithms and tools are fully agnostic to experiment-specific features, such as detector design or reconstruction framework details, allowing ACTS to be used across a variety of HEP experiments. In addition to its highly flexible and customizable design, ACTS provides an extendable R&D platform for the development and study of novel techniques and algorithms, such as machine learning-based track- and vertex reconstruction approaches.

6.1.2 Conceptual Design

ACTS is designed to provide a modern software toolkit that can be directly integrated in experiment-specific computing frameworks with a high flexibility in easily extending and adapting its functionality to specific experiment's needs. A high-level event data model (EDM) as well as generalized algorithmic formulations aim to guarantee a transparent and seamless utilization of the provided software modules.

In the recent years, many different HEP data processing frameworks have been prepared to make most efficient use of the expected limited available computing resources in future high-luminosity collider conditions by adapting them to multi-threading (MT) computing workflows, where multiple execution threads can run concurrently while potentially sharing common resources. All ACTS software components are therefore inherently designed to be used across different threads in an MT-environment without causing concurrency problems, i.e. the provided modules are designed to be *thread-safe*.
In order to meet the above described design expectations, all implemented ACTS modules follow a set of central coding concepts that are summarized in the following:

- An abstraction from specific experimental details is accomplished by a generic EDM and geometry description.
- Algorithmic implementations are given in general formulations that are not dependent on any experiment-specific aspects.
- The core components have only minimal dependency on external software or libraries, allowing an easy utilization and integration. The only two required dependencies are the `Eigen` [119] math library for linear algebra as well as the `Boost` [120] library for unit testing of the core components.
- All algorithmic modules are stateless engines without any knowledge of previous executions or configurations. If internal states are required, e.g. for performance-oriented information caching, the cached state must be provided by the caller (more details in an example below), in order to guarantee thread-safe behavior.
- The tool configuration is given by an encapsulated, customizable configuration struct, provided during tool initialization.
- Experiment-specific contextual data, such as the magnetic field status or detector calibration data, is transparently handled and passed through the full ACTS call structure, ensuring that all involved algorithms are fully aware of all given contextual aspects in the current call.
- Further extension of the provided algorithms can be achieved by a plugin mechanism for external software, e.g. including machine learning libraries.
- The software is written in modern C++ language, in a modular and well documented manner, facilitating future code maintainability and improvements.
- For performance reasons the usage of virtual inheritance (run time polymorphism) is minimized as much as possible. Instead, modularity and interchangeability of various software components is guaranteed by means of compile time polymorphism with extensive use of class and function templates.

An example implementation of a stateful ACTS algorithm with a nested configuration type is shown in Listing 6.1, illustrating the general design structure of any configurable, thread-safe algorithm implemented in ACTS.

The `Config` struct encapsulates all necessary algorithm configuration variables and a configuration object of this type is typically needed to construct an algorithm instance.
A `State` type object holds all state information of the given algorithm, i.e. intermediate or cached results and values. If a method requires cached information or possibly modifies the algorithm's state (see method `doSomething()`), the explicit state object must be provided by the caller during method invocation. In this way, the algorithm itself remains stateless and const-correct, i.e. no member function can modify any internal data or state of the algorithm it belongs to, making it a thread-safe quantity that is suitable for multi-threaded utilization.

```
// Example ACTS algorithm
class Algorithm {
public:
  // Nested configuration struct
  struct Config {
    int configVariableA = 0;
    double configVariableB = 0.;
  };

  // Nested state struct
  struct State
  {
    int cachedValue = 0;
  };

  // Construct the algorithm from its configuration
  Algorithm(const Config& cfg);

  // ‘const‘ method requiring or modifying cached information
  void doSomething(State& state) const;

private:
  // The configuration object
  Config m_cfg;
};
```

Listing 6.1: Example ACTS algorithm with a nested configuration type as well as an algorithm state. All cached information is contained in the state type and the algorithm itself remains stateless and const-correct.

6.1.3 Core Components

ACTS organizes all of its algorithms and tools in different modules, grouping software components of similar functionality together.
An `EventData` module defines the underlying EDM, including classes for e.g. track parameters or single measurement descriptions, which is used for intra- and inter-communication by all implemented software components. Detector geometry and magnetic field descriptions are handled by the `Geometry` and `MagneticField` modules, respectively. The `Surfaces` module implements various surface types used for detector and boundary surface descriptions, while surface- and volume-based materials are described by the `Material` components. Track finding and track fitting functionality is given by the `TrackFinding` and `TrackFitting` modules, respectively, whereas the `Propagator` module provides tools

for particle state propagation and covariance matrix transports along their respective trajectories in different magnetic field configurations.
The newly added `Vertexing` module functions mostly as a standalone group of vertex reconstruction components that internally depend on some of the other available ACTS modules, such as the propagation or magnetic field algorithms. A detailed overview of the ACTS vertexing module will be given in the following.

6.2 The ACTS Vertexing Suite

Based on the primary vertex reconstruction algorithms implemented in ATLAS and following the above described ACTS coding concepts, a modern, fast and inherently thread-safe vertex reconstruction software suite has been developed and implemented within the ACTS toolkit. Its goal is to provide robust, flexible and highly performant vertexing algorithms that can easily be integrated in different experiment-specific software frameworks and customized to their specific needs. Since the ultimate tuning to achieve excellent *physics* performance is highly experiment-specific, and will therefore be subject to physics optimization studies performed by the software's client, the main focus of the software implementation will lie on its *computational* performance.
In the following, an overview of all included components as well as their interplay will be given. Implementation details as well as various performance optimization aspects will be discussed in Section 6.3.

6.2.1 Overview of Vertexing Components

The ACTS vertexing software suite comprises numerous different vertex reconstruction algorithms and tools, which can be grouped into several categories. The most important categories include the vertex finder, vertex seed finder as well as vertex fitting algorithms, as they represent the main building blocks of nearly any vertex reconstruction chain. They are complemented by auxiliary tools such as different track linearization methods, a variety of more general vertexing-related utilities and a dedicated vertexing event data model.
Fig. 6.1 diagrammatically illustrates the various implemented ACTS vertexing components as well as their interplay among each other and with other ACTS modules. Two different vertex finding algorithms (green), an Adaptive Multi-Vertex Finder (AMVF) as well as an Iterative Vertex Finder (IVF), implement both a finding-through-fitting as well as a fitting-after-finding approach, as described in Section 5.2. They internally depend on vertex fitting (light blue) as well as seed finding methods (orange). Available vertex fitters include an Adaptive Multi-Vertex Fitter as well as a Billoir Vertex Fitter algorithm while the seed finding components comprise both a Gaussian Seeder and a Z-Scan Seeder as well as the two newly developed seeding methods presented in Section 5.4, the Grid Seeder and the Adaptive Grid Seeder.
Vertex fitting methods require knowledge about linearized track states which can be provided in ACTS by the track linearization components (blue). Note that the fundamental concepts of a semi-numerical Generic Track Linearizer as discussed in Section 5.3.5 have

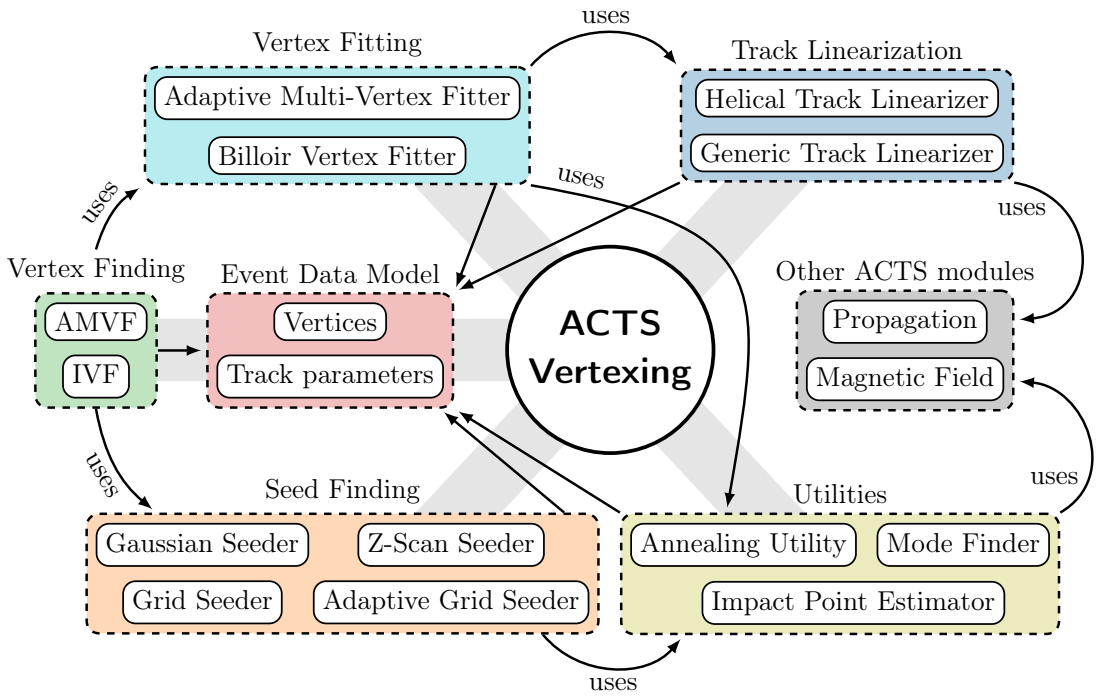


Figure 6.1: Schematic overview of all implemented vertex reconstruction components in the ACTS vertexing module, together with their interdependencies illustrated by arrows. All vertexing components rely on the common, underlying vertexing event data model (as indicated by the arrows without labels).

been developed and introduced in the ACTS vertexing module, however, as some key functionality required to be provided by the ACTS Propagation module is not yet available at the time of writing, the Helical Track Linearizer is the only current production-ready track linearization tool in ACTS.

Additionally, both vertex fitting as well as vertex seed finding methods internally employ various dedicated utilities (yellow), such as tools for impact point estimation, mode finding or simulated annealing.

The dependency on other ACTS modules (gray), in particular the ACTS Propagation and Magnetic Field modules, is given by both the track linearizers and vertexing utilities being reliant on particle state propagation and magnetic field information.

All vertexing components depend on a dedicated underlying time-dependent event data model which was specifically developed for the ACTS vertexing module. A detailed description is given in Appendix A.1.

Furthermore, maximal flexibility is guaranteed by means of compile-time polymorphism, which allows to interchangeably use different tools or algorithms with similar functionality that share the exact same interface. In practice, this permits a user to easily switch between e.g. different underlying seed finding or track linearization algorithms without the need of changing any class interfaces. Details on the available public interfaces and the usage of class templates is provided in Appendix A.2.

Since one of the main goals of the ACTS software toolkit is to provide a set of experiment-independent track- and vertex-reconstruction tools, the implementation of the vertexing software must allow a flexible, robust and performant way of utilizing arbitrary *user-defined* track types as input to the vertexing algorithms. Thus, user must not be bound to a pre-defined ACTS-specific track type, but instead should be able to run the vertex reconstruction on any user-defined, experiment-specific track type. If, for instance, a user stores experiment-specific information about the track within the track object, such as e.g. a link to the underlying MC particle, and likes to reconstruct vertices from a set of these track objects while retaining all user-specific information also after the vertex reconstruction, the software must ensure a robust and efficient propagation of all user-defined information through the entire reconstruction process.
This is achieved by making all relevant ACTS vertexing components C++ template classes which are dependent on the input track type as a C++ template parameter, allowing the ACTS vertexing suite to handle basically any given input track type. A detailed description of how this feature was implemented is given in Appendix A.3.

6.3 CPU Performance Optimization of the ACTS Vertexing

This section discusses optimization studies for the per-thread CPU performance of the ACTS vertexing, in which performance bottlenecks in the original ATLAS implementation are identified and subsequently circumvented in the optimized ACTS implementation.
Note that this work solely focuses on CPU performance optimization studies, as opposed to physics performance optimizations, which are subject to studies performed by the experiments employing the software in their reconstruction frameworks, as the algorithm's fine-tuning for optimal physics results is a highly experiment-specific task.
Optimal CPU performance, on the other hand, must be guaranteed from the ACTS software side, independent of the actual use case and application.
Several optimization strategies have been developed and deployed to identify and diminish CPU performance bottlenecks in the implementation of the ACTS vertexing algorithms. Special focus was put on *not* affecting the algorithmic behavior of the algorithms and tools, i.e. the physics output remains entirely unchanged, while the desired CPU time speed-ups exclusively stem from C++ code optimization strategies such as the application of optimized data structures, caching of CPU expensive mathematical computations or the utilization of dedicated efficient C++ features.
The following studies are performed analyzing and optimizing specifically the CPU performance of the Adaptive Multi-Vertex Finder (together with the Gaussian Seeder as its vertex seed finding algorithm), as it will play a major role in the primary vertex reconstruction in future high-luminosity environments, such as the ATLAS primary vertexing in LHC Run 3 and beyond. While certain optimization aspects are very specific to the algorithmic behavior of the AMVF, and will therefore specifically speed up this particular algorithm, several more general CPU performance-optimizing improvements, such as the utilization of efficient data structures, have been derived from these studies and applied across the entire ACTS vertexing module.

The CPU performance optimization was performed in several subsequent stages: prior to the actual implementation of the ACTS vertexing algorithms, the CPU performances of their original implementations in ATLAS were studied and performance bottlenecks identified, allowing the ACTS vertexing software to be directly implemented in a performance-oriented manner, overcoming several inefficiencies from the very beginning. In a second and third stage of the CPU performance optimization studies, the implemented ACTS vertexing software was further profiled and revised, resulting in a fast and highly optimized vertexing software suite.
For the sake of conciseness, all optimization details will be discussed together in the following without differentiating between pre- and post-implementation optimization. Therefore, the original ATLAS AMVF implementation will be used as the pre-optimization reference for demonstrating inefficiencies and bottlenecks to be avoided and improved upon in the ACTS implementation.

6.3.1 Reference Performance – The Original ATLAS Implementation

The CPU performance of the pre-optimization AMVF reference implementation in ATLAS was tested and profiled on 3000 $t\bar{t}$ events with a $\langle\mu\rangle = 40 - 60$ distribution using a machine equipped with two 3.30GHz Intel(R) Xeon(R) CPU E5-2667 v2 8-core processors, 32 GB of RAM and the CERN CENTOS 7 operating system. On average, a total primary vertex reconstruction time per event of 260 ms is observed, where the main contributions are coming from the vertex seed finding components (50.5%), followed by the vertex fitting (34.8%) as well as other utilities (14.7%). Fig. 6.2 shows these relative contributions to the total average reconstruction time, which are in turn subdivided into major contributing sub-algorithms and function calls.
The by far computationally most expensive part of the entire primary vertex reconstruction chain is the evaluation of track densities in the Gaussian vertex seed finding algorithm, manifested in large contributions from exponential function calls (29.6%) and memory access in the iteration over maps storing track density information (13%). While one part of the vertex seeding CPU time arises from algorithmic inefficiencies of the Gaussian seeder itself and cannot be directly optimized — again emphasizing the need of the newly developed (Adaptive) Grid Seeder in Section 5.4.3 and 5.4.4 — the other major contribution stems from the deployment of inappropriate data structures with slow memory access as well as avoidable recalculations of cacheable mathematical expressions, both of which can be optimized to speed up the vertex seeding execution time.
Other significant contributions to the total CPU time arise from the vertex fitting side, where Kalman vertex position updates (10.1%) involve avoidable recalculations of expensive matrix operations, such as matrix inversions, or the calculation of annealing weights (8.7%) which, again, comprise many evaluations of exponential functions.
The optimization studies performed to improve upon the performance of the above given reference ATLAS AMVF implementation involved some major optimization steps, such as the deployment of more efficient data structures, as well as possibly hundreds of various small performance-oriented code modifications throughout the entire ACTS vertexing

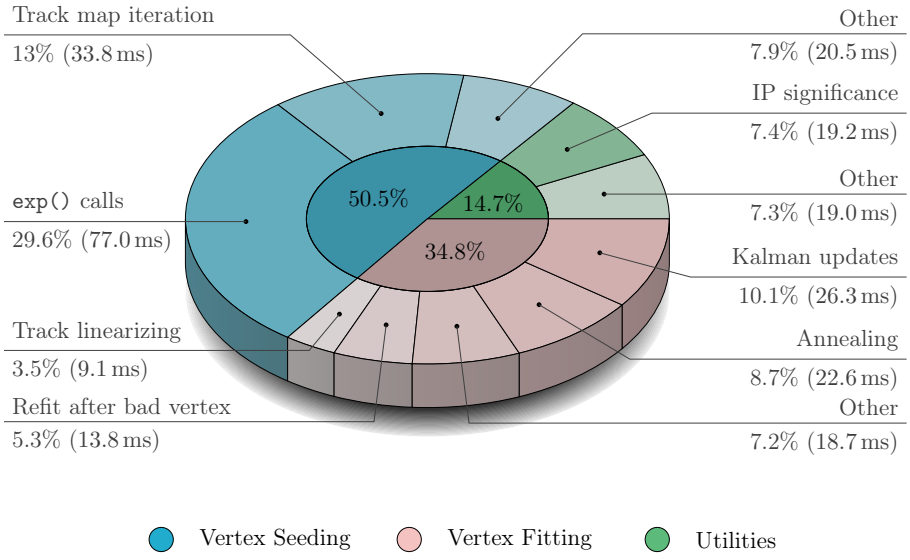


Figure 6.2: Relative contributions of different sub-algorithms to the total primary vertex reconstruction time by the ATLAS reference AMVF algorithm, grouped into three major contributing categories, vertex seeding, vertex fitting and other utilities. All displayed numbers are averaged values for single-threaded execution on 3000 $t\bar{t}$ events with a $\langle\mu\rangle = 40 - 60$ profile. The total average reconstruction time amounts to 260 ms with vertex seeding, vertex fitting and other utilities contributing 131.3 ms, 90.5 ms and 38.2 ms, respectively.

software that ultimately add up to make the code base more efficient. As not all of them can be listed and explained for the sake of conciseness, the following only summaries the key optimization concepts by discussing selected exemplary code improvements that have been applied in similar ways throughout the entire ACTS vertexing software.

6.3.2 CPU Performance Optimization Strategies

Utilization of Efficient Data Structures

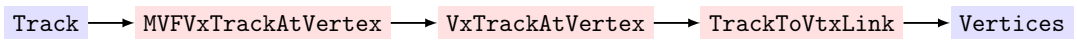
Primary vertex reconstruction generally involves a large number of modifications of data containers holding track related information, such as insertions of track objects to a list of tracks being compatible with the current vertex candidate or deletions of tracks from the seed track collection after they have been associated to a fitted vertex. In any case, insertions and deletions of elements in a data container entails (re-) allocation of memory, which can become computationally very expensive, as often all elements in the data container need to be copied to a new location in memory (e.g. if the data is internally stored contiguously and the current block of memory is not sufficiently large to hold the newly added element). Therefore, the most efficient way of handling these (unavoidable) data container modifications is to store light-weight objects, such as pointers, in the containers whose (re-) allocation is computationally less expensive. The entire ACTS vertexing suite is thus based on raw pointers to track objects, rather than the track objects themselves. The tracks used for primary vertex reconstruction in ACTS are therefore hold as `const` objects outside of the vertexing algorithms, while they are internally represented as pointers that can be handled and dealt with efficiently.

For the same reason, vertex objects are created at the vertex seeding stage and subsequently only vertex pointer objects, pointing to the vertex object of interested, are passed around by the vertex finder algorithm to all its associated utilities. All operations performed on vertices, such as position updates or the association of new tracks, are thus executed using pointers to the initial seed vertices without the need of copying the actual vertex objects.

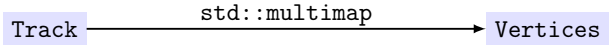
Additionally, the optimized ACTS vertexing implementation focuses on the utilization of appropriate data containers that are most efficient in their respective use case: around 13% of the total primary vertex reconstruction time of the pre-optimization AMVF reference implementation is caused by the increment operator in the iteration over an `std::map` storing track density information for the Gaussian seed finder. So-called associative containers, i.e. containers such as the `std::map`, implement sorted data structures in C++ that can be quickly searched but are not well suited for sequential iterations, as their elements are *not* stored contiguously in memory. Hence, the application of inappropriate data containers can cause massive overheads in frequently invoked execution paths. Data structures in the ACTS vertexing are adapted in such way that they can be stored in the most efficient data container for their respective application. In the above mentioned example of an `std::map` storing track density information, the code was restructured such that a newly introduced data type `TrackEntry` acts as a wrapper for all track density related information, which can be stored in an `std::vector`, the perfect and most efficient data container for sequential iterations.

Furthermore, special focus is put on the implementation of shallow call and data structures to avoid unnecessary performance overheads. Fig. 6.3 exemplarily shows the call structures needed during a multi-vertex fit, when a list of all vertices that currently use a specific track needs to be retrieved, comparing the original ATLAS and optimized ACTS implementations.

While the original AMVF implementation in ATLAS relies on several nested stages to get from the track object to the list of vertices currently using this track, and therefore inducing a performance overhead, the ACTS implementation makes use of the C++ provided `std::multimap` container which allows to efficiently associate a single key (i.e. a track pointer) to multiple values (i.e. pointers to all vertices currently making use of the track).



(a) ATLAS call structure



(b) ACTS call structure

Figure 6.3: Exemplary comparison of the call structures needed for track-to-vertices association during a multi-vertex fit. While the deep ATLAS call structure (a) involves many separate steps with nested data types, the ACTS track-to-vertices association (b) is based on an efficient and shallow implementation using an `std::multimap`.

Efficient data structures and containers like the ones exemplarily described above are employed throughout the entire ACTS vertexing code base in order to guarantee minimum overhead and maximum CPU performance.

Caching of Computationally Expensive Expressions

Caching computationally expensive mathematical expressions, and therefore avoiding to recalculate them in multiple different places, can also help to speed up the algorithms' CPU performances. Several internal, thread-local cache objects have been introduced in many ACTS vertexing tools and algorithms that can be used to pass cached information around, eliminating performance overheads from the recalculations of already previously derived quantities. An example is the implementation of the `MatrixCache` object that is used to cache matrix inversions and other computationally expensive matrix operations that are needed in several different places during the Kalman vertex update step.
Repeated evaluations of trigonometric functions, exponential function or even divisions in hot execution paths with several millions of invocations can furthermore have a significant performance impact and are therefore reduced as much as possible. Examples include, among many others, the `std::exp` calls in the Gaussian seeder, which can, to a certain extend, be cached for the subsequent track density derivative calculations or divisions in the computation of track weights during the deterministic annealing. Here, expressions of the form $\frac{\hat{\chi}^2}{2T}$ (see e.g. Eq. (5.24)) need to be computed many million times for fixed temperature values T and can therefore be simplified to the computationally less expensive form $a\hat{\chi}^2$ (multiplications are generally less expensive than divisions) with the pre-computed factor $a = \frac{1}{2T}$ that only needs to be derived once.

Utilization of Efficient C++ Features

As already discussed in Section 6.1.2, one of the main ACTS coding concepts is the utilization of compile time polymorphism, i.e. class and function templates, instead of polymorphism at run time. In this way, potentially CPU expensive virtual table[1] lookups at run time can be avoided and instead, the resolution of function calls is already performed at compile time, resulting in an overall more efficient CPU performance.
Furthermore, whenever possible dedicated and efficient C++ standard template library algorithms are employed instead of raw loops when operations on elements in data containers are needed. This enhances the algorithmic performances as well as code readability and maintainability.
Additionally, certain C++ properties, such as the so-called short-circuit evaluation, in which seemingly very small code changes can have a significant performance impact, are exploited throughout the entire ACTS vertexing suite. Short-circuit evaluation refers to the early termination of the evaluation of a logical expressions if its first argument does not satisfy the required condition: if, for instance, the first argument of the logic `AND` (`&&`) operation evaluates to `false`, which will result in an overall `false` outcome independent on the

[1] A so-called virtual table in C++ is a lookup table used to resolve virtual function calls at run time in a dynamic binding fashion.

value of the second argument, the evaluation of the second argument will be skipped, as exemplarily demonstrated in Listing 6.2.

```
/* Consider conditionA = true and conditionB = false */
// Both conditions will be checked
if (checkConditionA() && checkConditionB()){
  doSomething();
}
// C++ short-circuit evaluation:
// checkConditionA() function will not be called
if (checkConditionB() && checkConditionA()){
  doSomething();
}
```

Listing 6.2: Example illustrating the C++ short-circuit evaluation property.

This feature is leveraged in many places within the ACTS vertexing software, where the evaluation of multiple nested conditions is required. If the evaluation of condition A (i.e. the function `checkConditionA()` in the example listing) is computationally much more expensive than the evaluation of condition B, the order of evaluation can strongly influence the resulting CPU performance. About 7.4% of the total ATLAS reference AMVF CPU time is spent in the calculation of track IP significance values (see Fig. 6.2), a calculation that is triggered within a nested control flow statement similar to the one exemplarily shown in Listing 6.3, Example 1. Since the second condition evaluation (`checkZdistanceCondtion()`) is computationally much less expensive than the first one, a very simple change of the evaluation order can result in a significance performance improvement, since possibly thousands of expensive function calls can be easily omitted.
The ACTS vertexing implementation makes use of these subtle, yet very effective implementation details in many different places with the goal of providing a highly optimized CPU performance.

```
// Example 1: ATLAS-like implementation
if (checkIPSignificanceCondition() && checkZdistanceCondtion()){
  doSomething();
}

// Example 2: ACTS implementation
// More efficient by virtue of short-circuiting
if (checkZdistanceCondtion() && checkIPSignificanceCondition()){
  doSomething();
}
```

Listing 6.3: Exemplary comparison of nested control flow statement handling between the pre- and post-optimization implementation. Short-circuit evaluation is leveraged to avoid CPU expensive condition checks whenever possible.

Further Optimization Studies – Annealing and Vertex Refits

In addition to the optimization methods discussed above, two more potential CPU performance improvements have been identified, but were eventually *not* implemented in the ACTS vertexing suite for reasons described in the following.

The CPU time of the vertex fitting process could be further improved by reducing the number of exponential function calls in the determination of track weights. According to Eq. (5.24), the weight function in an adaptive multi-vertex fit is given by

$$\omega_{ij}(\hat{\chi}^2_{i1},\ldots,\hat{\chi}^2_{iM},T) = \frac{\exp\left(-\frac{\hat{\chi}^2_{ij}}{2T}\right)}{\exp\left(-\frac{\hat{\chi}^2_{\text{cut}}}{2T}\right)+\sum_{k=1}^{M}\exp\left(-\frac{\hat{\chi}^2_{ik}}{2T}\right)}. \tag{6.1}$$

By virtue of exponential laws, the equation can be rewritten as

$$\omega_{ij}(\hat{\chi}^2_{i1},\ldots,\hat{\chi}^2_{iM},T) = \frac{1}{\exp\left(\frac{1}{2T}(\hat{\chi}^2_{ij}-\hat{\chi}^2_{\text{cut}})\right)+\sum_{k=1}^{M}\exp\left(\frac{1}{2T}(\hat{\chi}^2_{ij}-\hat{\chi}^2_{ik})\right)}, \tag{6.2}$$

reducing the number of exponential function calls by one, for every track weight calculation. However, it was observed that in case of a very large $\hat{\chi}^2_{ij}$ value, its positive contribution in the exponential functions in the weight's denominator can cause a floating point overflow, i.e. the resulting function value is too large to be represented by a floating point variable. In the non-optimized representation in Eq. (6.1), $\hat{\chi}^2_{ij}$ contributes negatively to the exponential function, resulting in a very small function value in the numerator for a large $\hat{\chi}^2_{ij}$, as opposed to a very large value in the denominator of the optimized version, and can therefore not cause any floating point overflow issues.

Another potential performance improvement could entirely eliminate the 5.3% CPU time contribution (see Fig. 6.2) from refits after a bad vertex. A vertex for which no compatible tracks remain after the multi-vertex fit has been performed is referred to as a *bad* vertex and will be removed from the collection of fitted vertices by the AMVF. As all other, previously found vertices have been refitted in a multi-vertex fit including the bad (and now again removed) vertex, a full refit excluding the bad vertex has to be performed after vertex removal.

An option to cache the previous vertex fitter state, including all necessary information about the fitter and all vertices before the bad vertex was added, to which the fitter can return in case a vertex has to be removed from the fit, was added to the ACTS AMVF implementation. Indeed, without the need of rerunning a full multi-vertex fit every time a bad vertex is removed from the fit and just returning to the previous state of the fit before the bad vertex was added, the expected $\sim 5\%$ performance gain could be seen. However, a small degradation in physics performance was observed, as these additional multi-vertex fits after the removal of a bad vertex — although not explicitly foreseen in the original formulation of the AMVF algorithm — seem to improve the overall fit convergence and thus its performance. As even small changes in physics performance are not desired in these

purely CPU performance oriented studies, this potential CPU performance improvement was not further considered in the performed optimization studies.

6.3.3 Optimized Performance of the ACTS Vertexing

The CPU performance of the post-optimization ACTS AMVF implementation was evaluated under the exact same conditions (i.e. using the same data set and machine in single-threaded execution mode) as the pre-optimization reference AMVF implementation analyzed in Fig. 6.2.

The developed optimization strategies implemented in the ACTS vertexing algorithms lead to a significant CPU performance enhancement, as shown in Fig. 6.4. The average CPU time spent for vertex seeding and vertex fitting on 3000 $t\bar{t}$ events is reduced by more than 50% whereas a more than 25% CPU time reduction can be seen for other utilities. The total average primary vertexing reconstruction time decreases accordingly by more than 50% from 260 ms (pre-optimization, ATLAS) to 125 ms (post-optimization, ACTS).

Fig. 6.5 shows the relative and total contributions of several sub-algorithms and function calls to the total average reconstruction time for the optimized ACTS AMVF implementation. The relative contributions of vertex seeding, vertex fitting and other utilities are

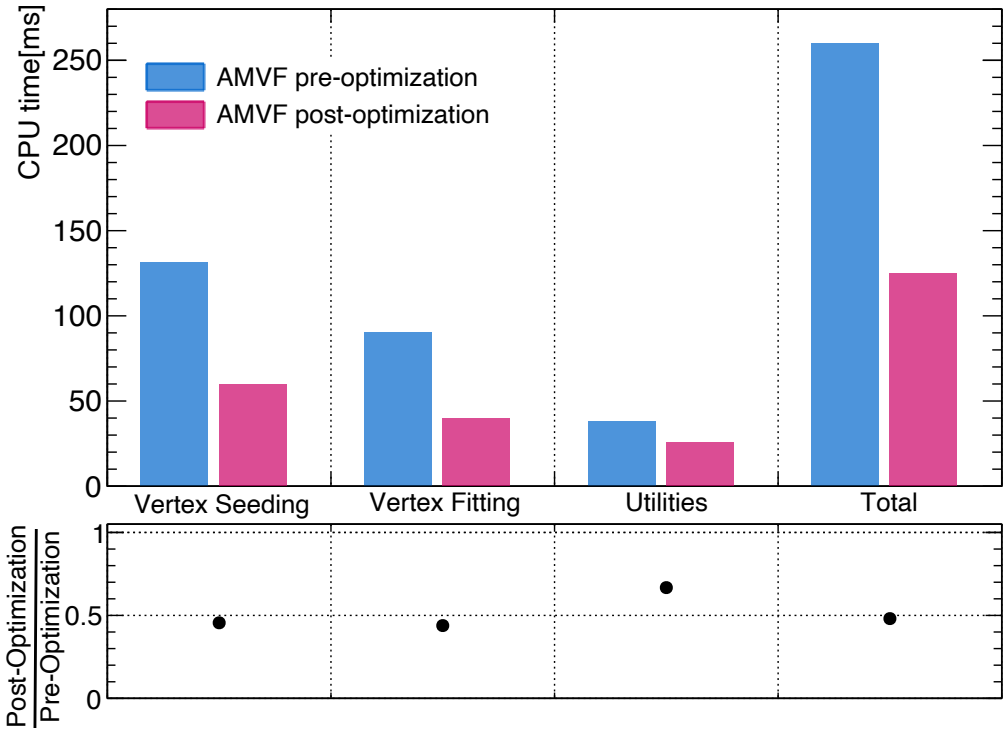


Figure 6.4: Average AMVF primary vertex reconstruction times for 3000 $t\bar{t}$ events with a $\langle\mu\rangle = 40 - 60$ profile in single-threaded execution mode, comparing the pre-optimization ATLAS reference and post-optimization ACTS implementation. CPU times for vertex seeding, vertex fitting and other utilities are shown individually in addition to their cumulative total average primary vertex reconstruction times.

comparable to the non-optimized reference implementation, while significant improvements of the total CPU time are observed in all shown sub-algorithms.
The utilization of efficient data structures and data containers immensely reduces the time spent for memory access in iterations of data collections, reducing for instance the CPU time for track iterations during vertex seeding from 33.3 ms to 2.2 ms. The efficient usage of dedicated C++ features as well as caching of mathematical computations results in significant performance gains throughout all employed algorithms and functions.
Even after optimization, more than 38% of the total primary vertexing CPU time is spent in irreducible exponential function calls needed for track density calculations of the Gaussian seed finding algorithm, once again motivating the newly developed highly performant (Adaptive) Grid Seeder algorithms in Section 5.4.3 and 5.4.4.

Significant CPU time improvements compared to the ATLAS reference implementation are achieved by applying dedicated optimization concepts to the newly implemented ACTS vertexing software, while the exact same physics output is to be expected as no algorithmic changes were made.
In-depth CPU and physics performance studies of the ACTS AMVF algorithm (studying both the utilization of the Gaussian Seeder as well as the new (Adaptive) Grid Seeder) and the ACTS IVF algorithm will be presented in Chapter 7 in the context of an ACTS-standalone application as well as the ACTS vertexing integration in the ATLAS software framework Athena.

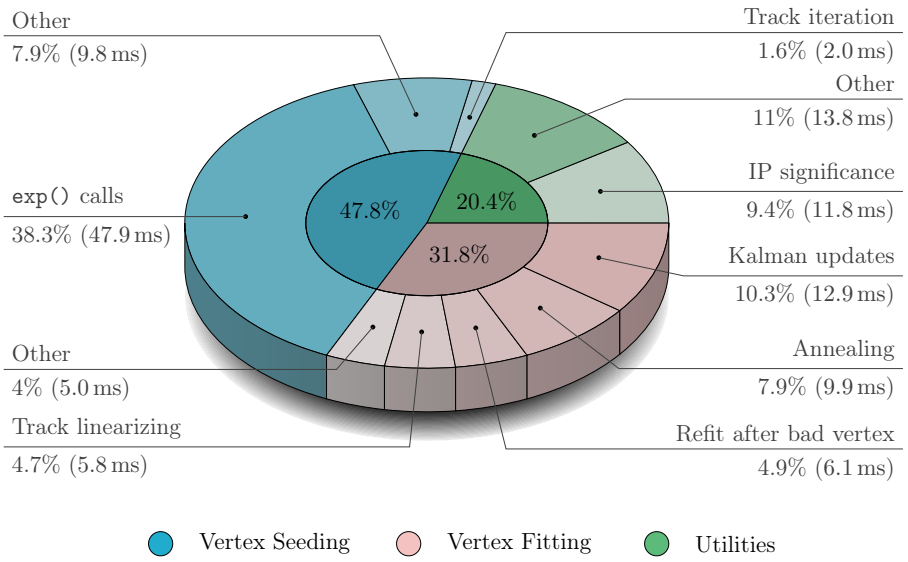


Figure 6.5: Relative contributions of different sub-algorithms to the total primary vertex reconstruction time by the optimized ACTS AMVF implementation, grouped into three major contributing categories, vertex seeding, vertex fitting and other utilities. All displayed numbers are averaged values for single-threaded execution on 3000 $t\bar{t}$ events with a $\langle\mu\rangle = 40 - 60$ profile. The total average reconstruction time amounts to 125 ms with vertex seeding, vertex fitting and other utilities contributing 59.8 ms, 39.7 ms and 25.5 ms, respectively.

Chapter 7

Application and Performance of the ACTS Vertexing Software

The implemented ACTS vertexing software presented in Chapter 6 has been successfully integrated, tested and validated in two different software frameworks, a lightweight ACTS-internal event processing framework called ACTS-*Examples*, which provides standalone simulation and reconstruction examples for the usage of the ACTS core library modules, as well as the ATLAS software framework Athena [81].
This chapter first presents the physics and CPU performances of the ACTS vertex finder algorithms in the context of a full reconstruction chain from event generation over fast track simulation and truth-based track fitting to the final primary vertex reconstruction in the ACTS-Examples framework (Section 7.1), followed by performance studies of the newly developed Adaptive Grid Seeder and Grid Seeder algorithms in the same ACTS standalone context (Section 7.2).
Section 7.3 then discusses the application and performance of the ACTS vertex reconstruction software in ATLAS. The original single-threaded design of the ATLAS software framework Athena was recognized to become increasingly problematic for the expected computational demands in the upcoming LHC conditions, leading to recent major software upgrades of the Athena framework to prepare for the HL-LHC era, where multi-threaded event processing will be inevitable.
In the context of this thesis, the ACTS vertexing software has been fully integrated, deployed and validated in the ATLAS software framework to replace the original ATLAS primary vertexing software and allow for MT-capable primary vertex reconstruction in ATLAS for LHC Run 3 and beyond. The performance of the ACTS vertexing deployed in ATLAS will therefore be discussed in both single- and multi-threaded execution mode, while also the application of the Grid Seeder algorithm is studied. The technical work that has been done to integrate the ACTS vertexing software in ATLAS and to prepare all other required ATLAS vertexing tools and algorithms for multi-threaded utilization is presented in Appendix A.7.

7.1 Performance of the ACTS Vertex Finders in ACTS

The two ACTS primary vertex finder implementations, the AMVF and IVF, have been integrated in the ACTS-Examples framework, providing both example utilizations of the vertexing algorithms for new users that would like to integrate the ACTS vertexing in their software framework, as well as a testbed for studying and validating existing as well as novel algorithms. The AMVF and IVF implementations have been studied with respect to their physics and CPU performances in the ACTS-Examples framework, for which a full reconstruction chain, including several individual steps from event generation to the final primary vertex reconstruction, has been employed. These reconstruction chain stages are similar to the ones discussed for ATLAS in Section 3.4 and can be summarized as follows:

- **Event generation:** The first step in the reconstruction chain is again the simulation of the physics event. Here, the proton-proton collisions, including the hard-scatter event as well as a certain number of overlaid minimum bias events following a Poisson distribution with a pre-defined mean value, are described by the event generator `PYTHIA 8` [83], resulting in an output list of generated particles together with their momentum four-vectors.

- **Fast track simulation and digitization:** In the next step, the interactions of the generated particles with the detector material need to be simulated. Instead of a full detector simulation including detailed descriptions of the detector geometry and material distribution, a fast track simulation is deployed, in which a simplified detector geometry and material description is used to simulate the most relevant physics processes, such as stochastic multiple scattering, energy loss effects or decays of the particle traversing the detector. The ACTS-provided detector geometry used for the following studies was originally developed for the TrackML challenge [121] (and is therefore hereafter referred to as TrackML detector geometry), featuring an all-silicon typical LHC tracking detector layout with 10 separate layers of disks and cylinders. A constant magnetic field $B = (0, 0, 2\,\mathrm{T})$ is assumed in the simulation. For each hit produced by fast track simulation, the detector response is simulated using a fast digitization algorithm, resulting in one- or two-dimensional measurements (depending on the detector type) that are used as input to the subsequent track reconstruction.

- **Truth-based track fitting:** As no production-ready track finding algorithm is available in ACTS at the time of writing, the pattern recognition stage of the following track reconstruction is replaced by truth-based groupings of space points belonging to single particles. A Kalman filter is subsequently deployed to fit the track parameters, which are then passed to the final step of reconstruction chain, the primary vertex reconstruction.

- **Track selection and primary vertexing:** In order to mimic a realistic detector acceptance and to mitigate the influence of secondary tracks in the primary vertex

reconstruction, dedicated selection criteria are applied to the truth-based fitted track parameters:

- $|\eta| < 2.5$
- $p_T > 400\,\text{MeV}$
- $d_0 < 4\,\text{mm}$
- $q \neq 0$

The charged particle tracks that fulfill these selection criteria are subsequently passed to the primary vertex finder algorithm to find and reconstruct the primary interaction vertices. The fit is performed without the application of a beam spot or seed constraint. By means of retained links between the fitted track parameters and the original generated particles, the reconstructed vertices can be matched to their associated true vertex, allowing for efficiency and position resolution determination.

7.1.1 Physics Performance of the AMVF and IVF

In order to study the AMVF and IVF physics performance within the ACTS-Examples framework, the above discussed simulation and reconstruction procedure was utilized to produce 1000 $t\bar{t}$ events for each $\langle\mu\rangle \in \{20, 40, 60, 80, 100, 120, 140, 160\}$, resulting in a total of 8000 $t\bar{t}$ events covering a large range of pile-up scenarios, which are used in the following performance studies.

The AMVF and IVF algorithms are configured in an ATLAS Run 3- and ATLAS Run 2-like configuration, respectively. A detailed list of all configuration parameters for both algorithms is given in Appendix A.5.

Fig. 7.1 shows the average number of reconstructed primary vertices per event as a function of the true number of simultaneous pp collisions for both the ACTS AMVF and ACTS IVF algorithms. While the grey dashed line illustrates a hypothetical perfect reconstruction efficiency, the blue and purple dashed lines indicate the experimentally maximal achievable reconstruction efficiencies taking the detector acceptance as well as tracking efficiency into account, respectively. Although a truth-based track fitting is deployed, fit failures in the Kalman filter arising from e.g. bad initial parameter estimations can still result in a slight degradation of the maximal achievable efficiency, as observed in the given figure. For a true vertex to be considered reconstructable, at least two of its tracks must have been successfully fitted and passed the selection criteria.

While both AMVF and IVF are able to reconstruct most of the reconstructable primary vertices in low pile-up conditions with $\mu < 30$, a significant degradation in reconstruction efficiency is observed with the IVF for larger numbers of simultaneous pp collisions. At high pile-up values, the AMVF manages to resolve and reconstruct more than 50% of reconstructable vertices that are lost by the IVF.

As already discussed in Ref. [103], the drop in the IVF reconstruction efficiency for high μ can be explained by its tendency to merge close-by vertices, a very common scenario in high pile-up conditions, whereas the AMVF's superb ability to resolve close-by vertices

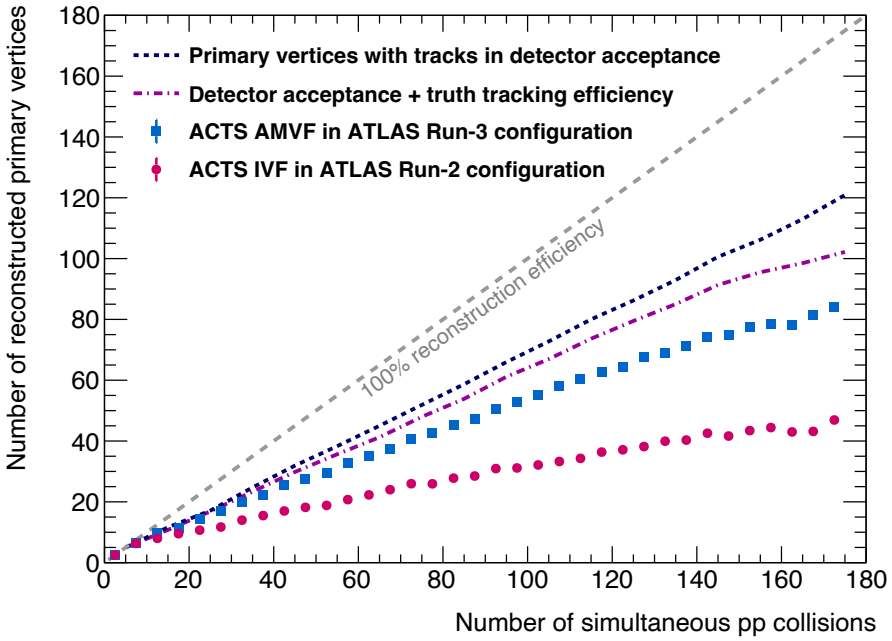


Figure 7.1: Average number of primary vertices reconstructed per event by the ACTS AMVF and IVF algorithms as a function of the true number of simultaneous *pp* collisions. For reference, a hypothetical 100% efficient reconstruction is indicated by a grey dashed line, while the other two dashed lines represent experimental reconstruction limits due to detector acceptance and tracking efficiency effects.

significantly enhances its overall performance in these challenging environments.
Fig. 7.2 shows the longitudinal separation of pairs of neighboring vertices reconstructed by the ACTS AMVF and ACTS IVF algorithms. A significant depletion around $\Delta z_{\text{closest}} = 0$ can be seen, which arises from vertices being merged and reconstructed as one single vertex if their longitudinal separation falls below a certain threshold. While the AMVF algorithm resolves and reconstructs separate vertices well below $|\Delta z_{\text{closest}}| < 1\,\text{mm}$, the IVF tends to merge vertices already at larger $|\Delta z_{\text{closest}}|$ values, resulting in the efficiency degradation in high pile-up environments discussed above.

The positions of the reconstructed vertices can be compared to the true vertex positions by matching reconstructed vertices with their corresponding true vertices. Using the retained links between reconstructed tracks and their associated truth particles, a reconstructed vertex can be matched to a true vertex: starting from the list of all tracks being attached to a certain reconstructed vertex, their associated truth particles can be determined. If a true vertex can be found from which at least 50% of the associated truth particles have originated, it is matched to the reconstructed vertex. Otherwise, multiple true vertices contribute significantly to the track content of the reconstructed vertex, for which reason the reconstructed vertex cannot be matched to one single true vertex.
Fig. 7.3 shows the z-position resolution of truth-matched vertices reconstructed by the ACTS AMVF and IVF algorithms. While the AMVF achieves a z-position resolution well below 0.1 mm, the IVF's distribution features a broader shape. The same behavior is

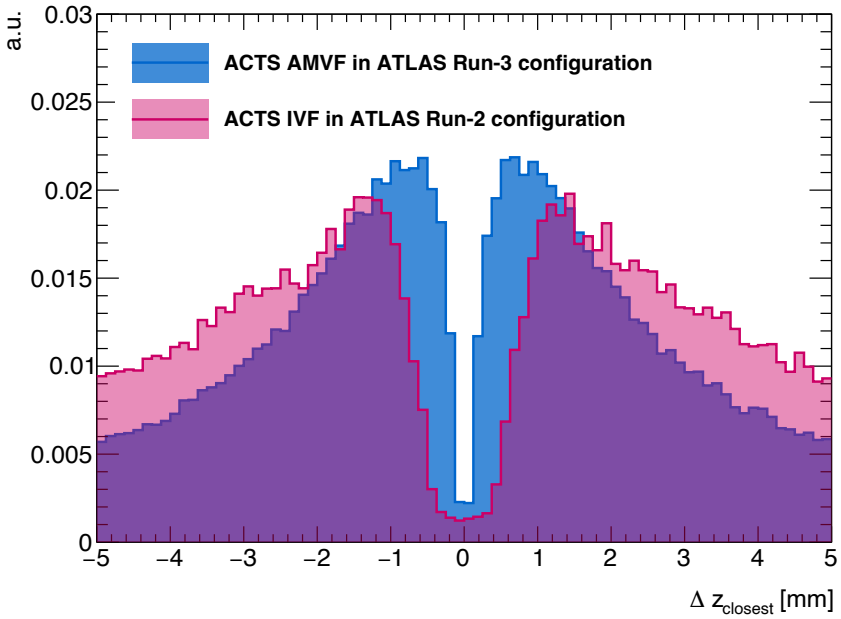


Figure 7.2: Longitudinal separation of pairs of neighboring vertices reconstructed by the ACTS AMVF and IVF algorithms. A depletion around $\Delta z_{\text{closest}} = 0$ arising from vertex merging can be observed.

observed in the x- and y-position resolutions shown in Fig. 7.4(a) and 7.4(b), respectively. For quantitative position resolution comparisons, means and standard deviations of the presented distribution, obtained by fitting the distributions with Gaussian functions, are shown in Table 7.1.

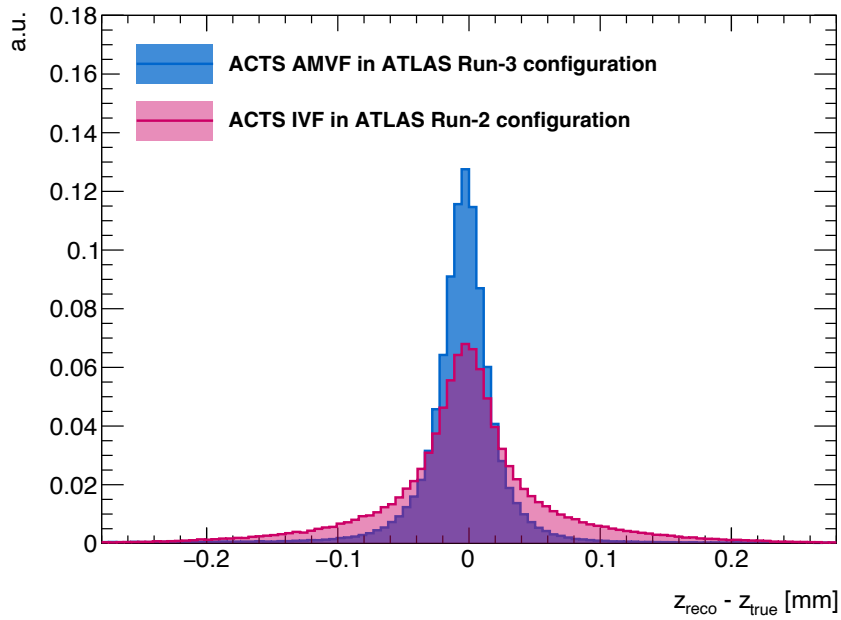
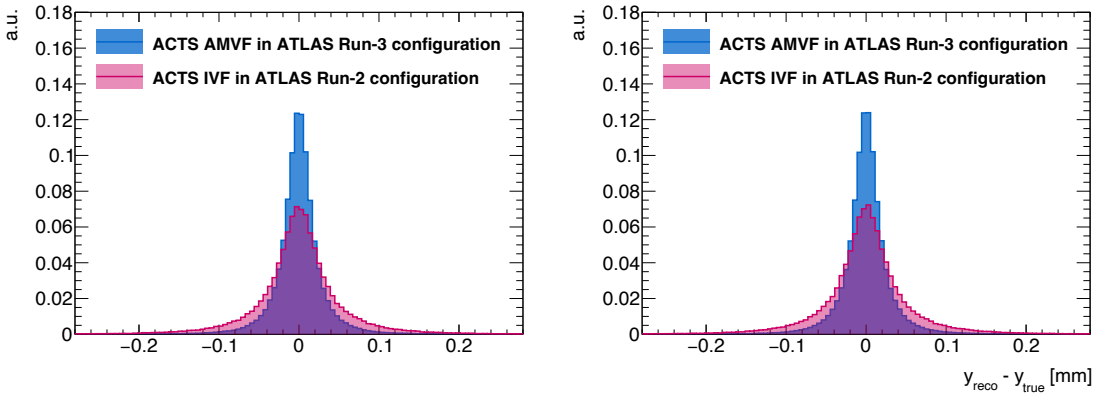


Figure 7.3: Resolution of z-positions of truth-matched vertices reconstructed by the ACTS AMVF and IVF. The truth-matching requires at least 50% of truth particles associated to the track content of the reconstructed vertex to have originated from a single true vertex.



(a) x-position resolution

(b) y-position resolution

Figure 7.4: Resolution of x- and y-positions of truth-matched vertices reconstructed by the ACTS AMVF and IVF. The truth-matching requires at least 50% of truth particles associated to the track content of the reconstructed vertex to have originated from a single true vertex.

Since fitted track parameters usually exhibit larger uncertainties in the longitudinal than in the transverse direction, an expected slight degradation in the longitudinal vertex position resolution compared to the transverse resolutions can be observed. Note that especially the absence of a beam spot constraint results in still comparable longitudinal and transverse resolutions, whereas an actual application of a beam spot constraint, as used in context of the ATLAS integration in Section 7.3, will generally significantly enhance the transverse resolutions.

	Algorithm	Mean [mm]	Standard deviation [mm]
x-position	AMVF	-1.4×10^{-5}	0.033
	IVF	-6.5×10^{-5}	0.061
y-position	AMVF	1.4×10^{-5}	0.033
	IVF	6.5×10^{-5}	0.060
z-position	AMVF	-5.8×10^{-3}	0.043
	IVF	-1.9×10^{-3}	0.071

Table 7.1: Mean and standard deviation values of the AMVF and IVF position resolution distributions.

7.1.2 CPU Performance of the AMVF and IVF

The CPU performances of the ACTS AMVF and IVF algorithms within the ACTS-Examples framework were compared on the same 8000 $t\bar{t}$ events data set described above, covering a wide pile-up range with $0 \leq \mu \leq 180$. All benchmark tests described in this section as well as in all following sections were run on a dedicated machine equipped with two 3.30 GHz Intel(R) Xeon(R) CPU E5-2667 v2 8-core processors, 32 GB of RAM and running the CERN CENTOS 7 operating system.

Fig. 7.5 shows the ACTS AMVF and IVF average primary vertex reconstruction times needed per event as a function of the number of simultaneous pp collisions per bunch

crossing. In low pile-up environments, where both algorithms performed almost equally well in terms of reconstruction efficiency, comparable CPU performances can be observed. While the average IVF reconstruction times scale almost linearly with the number of reconstructed vertices (compare Fig. 7.1), a significant, non-linear increase in reconstruction time for higher μ is seen for the more sophisticated AMVF algorithm. Especially for large numbers of simultaneous pp collisions, and therefore high track multiplicities, the irreducible performance impact of track density calculations in the Gaussian Seeder discussed in Section 6.3 leads to average primary vertex reconstruction times per event of well above 300 ms — a primary vertexing CPU performance that will be highly impractical for high luminosity environments.

The (Adaptive) Grid Seeder presented in Section 5.4.3 aims to provide a state-of-the-art vertex seed finding alternative that is specifically suited for high luminosity environments and mitigates the above mentioned CPU performance impact of the seed finding process to a bare minimum. Its physics and CPU performance in a standalone application as well as in the context of the AMVF algorithm will be presented in the next section.

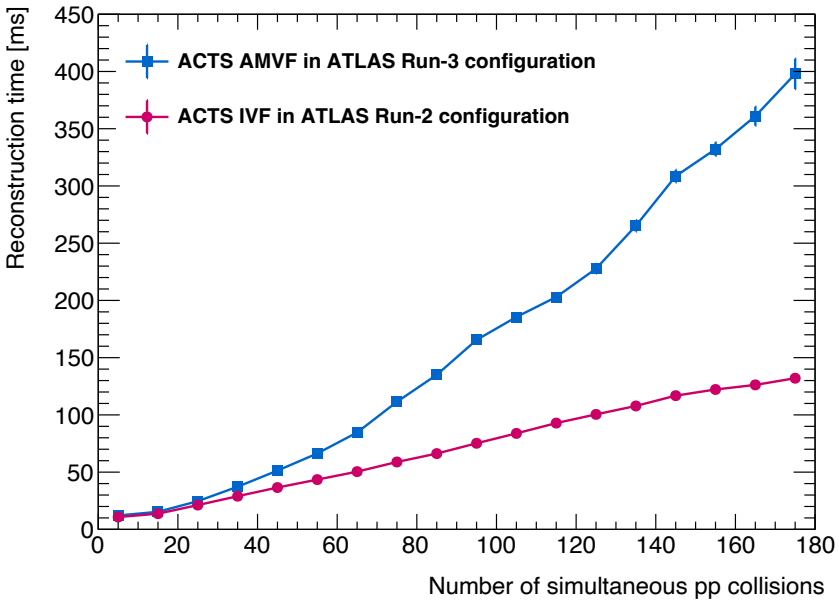


Figure 7.5: Average primary vertex reconstruction times needed per event as a function of the number of simultaneous pp collisions per bunch crossing for the ACTS AMVF and IVF in the context of the ACTS-Examples framework application.

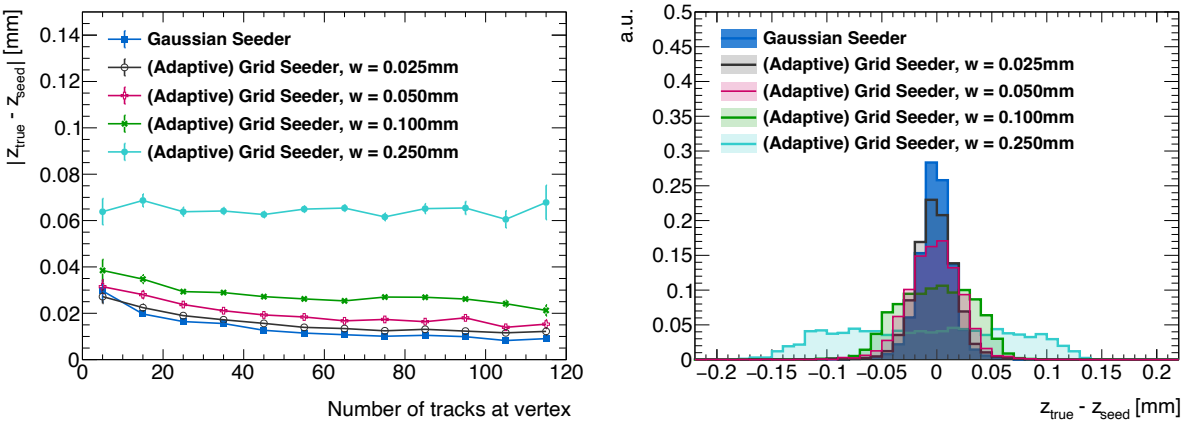
7.2 Performance of the (Adaptive) Grid Seeder in ACTS

This section discusses the physics as well as CPU performance of the non-adaptive and adaptive version of the Grid Seeder presented in Section 5.4.3 in the ACTS-Examples framework. Unless stated otherwise, the seed finding algorithms are deployed and tested on the same 8000 $t\bar{t}$ events used for studying the AMVF and IVF performances in Section 7.1.

7.2.1 Physics Performance of the (Adaptive) Grid Seeder

The physics performance of a vertex seed finding algorithm can only be fully understood in the context of a full vertex reconstruction chain, where every vertex that is not precisely found by the seed finder is likely to be lost and not reconstructed. However, in order to gain a better understanding of the behavior and functionality of the seed finding algorithm itself, an initial standalone analysis of the reconstruction resolution on isolated vertices can be advantageous. Thus, in the following, first the position resolution of the (Adaptive) Grid Seeder will be studied on isolated $t\bar{t}$ vertices, i.e. in the absence of any underlying pile-up vertices, and compared to the Gaussian seed finder. Afterwards, the seed finders will be examined in the full context of a primary vertex reconstruction chain, allowing to evaluate the total reconstruction efficiency performance and final vertex position resolutions of the full finding and fitting process in a variety of different pile-up scenarios.
Here, the non-adaptive and adaptive versions of the Grid Seeder can be studied together (and are therefore collectively referred to as (Adaptive) Grid Seeder or just Grid Seeder), as both algorithms only differ in the way they *construct* the track density distribution along the beam axis. Since the resulting density distribution is exactly the same for both algorithms (modulo entirely empty and unused bins in case of the non-adaptive version that do not change the physics outcome), their physics performances will be entirely identical.
For both versions, the most important configurable parameter with the largest physics performance impact is the chosen bin width w. Hence, different configurations of w, spanning from a very fine bin width of $w = 0.025\,\text{mm}$ (where the best physics results are expected) to an extremely coarse granularity with $w = 0.25\,\text{mm}$ or even $w = 0.5\,\text{mm}$ (where a practical utilization of the algorithm is not expected anymore) will be discussed. A detailed list of all configuration parameters used in the following studies can be found in Appendix A.6.
Fig. 7.6(a) shows the average absolute z-position resolution $|z_{\text{true}} - z_{\text{seed}}|$ between the true isolated $t\bar{t}$ vertices and the seed vertices found by the (Adaptive) Grid Seeder in different w-configurations and the Gaussian Seeder as a function of the number of input tracks to the vertex seed finding algorithms (i.e. the number of tracks at the isolated $t\bar{t}$ vertex). The Grid Seeders show the expected clear trend towards higher z-resolutions for more narrow bin widths. In case of a bin width of $w = 0.025\,\text{mm}$[1], the resulting seed z-position resolution is comparable with the one of the Gaussian seeder, whereas a very coarse bin width of $w = 0.25\,\text{mm}$ leads to a significant resolution degradation. This is expected behavior as the algorithm predicts the correct density bin (or rather its z-position center value) for the underlying seed vertex z-position but as the granularity is very coarse, the true vertex z-position can be thought of as a uniformly distributed random number drawn in the given bin interval. The standard deviation of such a uniformly distributed random number in an interval of width w is given by $\frac{w}{\sqrt{12}}$, a value consistent with the observed average absolute z-position resolutions for each given bin width w, respectively. The Gaussian Seeder as well

[1] A bin width of $w = 0.025\,\text{mm}$ corresponds to $N_{\text{bins}} = 20\,000$ density bins along the beam axis in case of the non-adaptive Grid Seeder with a z-coverage in the interval $[-250\,\text{mm}, 250\,\text{mm}]$. A detailed overview for all utilized w-configurations and their corresponding N_{bins} values can be found in Appendix A.6.



(a) (b)

Figure 7.6: (a) Average z-position resolution of the Gaussian and (Adaptive) Grid Seeders in four different bin width configurations on isolated $\mathrm{t\bar{t}}$ vertices as a function of the number of tracks at the vertex. (b) Overall seed vertex z-position resolution distributions of the Gaussian Seeder and the (Adaptive) Grid Seeder in the same w-configurations as in (a).

as all high granularity Grid Seeder configurations show an expected slight enhancement of seed vertex z-position resolution as more tracks, and therefore a potentially smoother track density distribution, is available.

The overall seed vertex z-position resolution distributions of the Gaussian Seeder as well as the (Adaptive) Grid Seeder in different w-configurations can be seen in Fig. 7.6(b). While the $w = 0.025\,\mathrm{mm}$ configuration of the Grid Seeder shows an almost comparable seed vertex position resolution to the one of the Gaussian Seeder, the flat and broad distribution with $(z_{\mathrm{true}} - z_{\mathrm{seed}})$ values in an interval of about $[-\frac{w}{2}, \frac{w}{2}]$ in the $w = 0.25\,\mathrm{mm}$ configuration is a manifestation of the extremely coarse granularity of the density bins.

However, the seed vertex z-position resolution is not a meaningful physics quantity in a standalone application and was presented above for illustrative reasons only. The most important feature of a vertex seed finding algorithm is its ability to actually *find* a vertex estimate with an accuracy that is suitable for a subsequent vertex fit, i.e. the meaningful physics quantity of interest is the total primary vertex reconstruction efficiency as well as the position resolutions of the final reconstructed and fitted vertex collection. In the following, the physics performance of the (Adaptive) Grid Seeder is therefore evaluated by employing it as the vertex seed finder in the context of a full primary vertex reconstruction chain performed by the AMVF.

Fig. 7.7 shows the average number of reconstructed primary vertices as a function of the number of simultaneous pp collisions for the AMVF algorithm employing the Gaussian Seeder as well as different configurations of the (Adaptive) Grid Seeder. Again, the grey dashed line illustrates a hypothetical perfect reconstruction efficiency, the blue and purple dashed lines indicate the experimentally maximal achievable reconstruction efficiencies taking the detector acceptance as well as tracking efficiency into account, respectively.

Additionally, the reconstruction efficiency ratio between different AMVF Grid Seeder configurations and the default AMVF configuration utilizing the Gaussian Seeder is shown. Although the rather coarse $w = 0.10\,\text{mm}$ configuration of the Grid Seeder led to an already sizeable degradation in seed vertex z-position resolution on isolated $t\bar{t}$ vertices in the standalone analysis above, its seed finding performance within the full vertex reconstruction chain is very much comparable to the Gaussian Seeder. This behavior reemphasizes the above mentioned importance of a seed finder's excellent vertex finding ability while a very accurate seed position resolution is not necessarily needed for the subsequent vertex fit in which the final vertex position will be fully determined.
The $w = 0.05\,\text{mm}$ version of the Grid Seeder shows a comparable, if not slightly better, physics performance compared to the $w = 0.10\,\text{mm}$ configuration in the context of the full reconstruction chain, with an efficiency ratio close to 1 throughout the entire analyzed pile-up range. As no significant enhancement in reconstruction efficiency could be observed by further decreasing the bin width from $w = 0.05\,\text{mm}$ to $w = 0.025\,\text{mm}$, the latter configuration is not explicitly shown in the given figure.

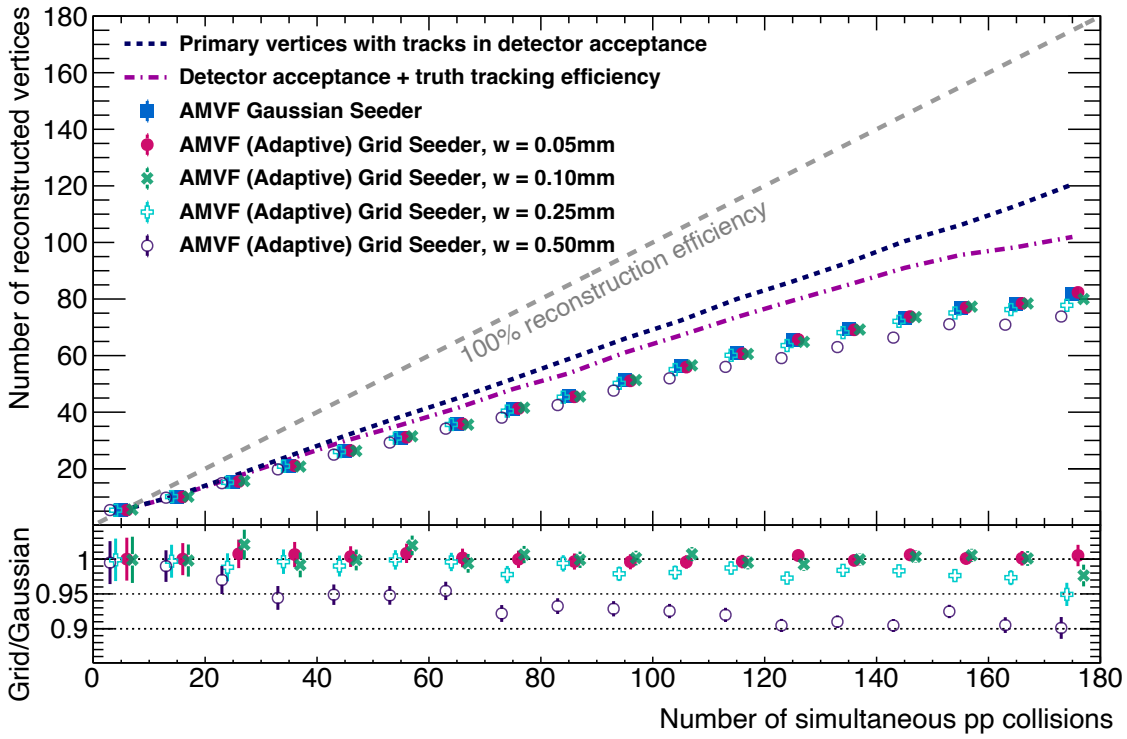
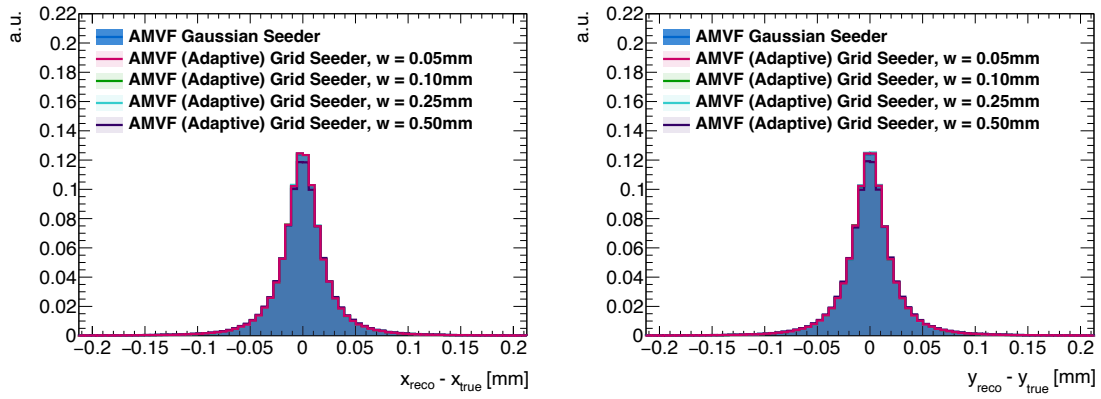


Figure 7.7: Average number of primary vertices reconstructed per event by the ACTS AMVF algorithm employing the Gaussian Seeder as well as different (Adaptive) Grid Seeder configurations as a function of the true number of simultaneous *pp* collisions. A small per-bin offset for the different entries is applied, in order to improve the visibility of overlapping entries. For reference, a hypothetical 100% efficient reconstruction is indicated by a grey dashed line, while the other two dashed lines represent experimental reconstruction limits due to detector acceptance and tracking efficiency effects.

Instead, in order to illustrate the Grid Seeder's behavior for extremely coarse bin widths, and therefore mark the lower end of the bin width spectrum in which excellent physics performance can be expected, the reconstruction efficiency of a $w = 0.50\,\text{mm}$ configuration[2] can be seen in addition to the $w = 0.25\,\text{mm}$ version already discussed in the standalone analysis on isolated $\text{t}\bar{\text{t}}$ vertices. While a bin width of $w = 0.25\,\text{mm}$ still results in reconstruction efficiencies only $1\% - 5\%$ below the ones of the Gaussian Seeder — a remarkable result considering the very significant seed position resolution degradation that was observed in the standalone analysis in this specific configuration — a clear efficiency drop is seen in the extreme case of $w = 0.50\,\text{mm}$.
The resulting transverse and longitudinal position resolutions of the final reconstructed and fitted vertices are compared in Fig. 7.8 and Fig. 7.9, respectively.



(a) x-position resolution

(b) y-position resolution

Figure 7.8: Resolution of x- and y-positions of truth-matched vertices reconstructed by the ACTS AMVF deploying the Gaussian Seeder as well as different (Adaptive) Grid Seeder configurations. The truth-matching requires at least 50% of truth particles associated to the track content of the reconstructed vertex to have originated from a single true vertex.

While a deployment of the high granularity configurations of the Grid Seeder leads to z-position resolutions very similar to the one of the Gaussian Seeder application in the AMVF, a considerable worsening including an induced bias starts to be apparent in the very extreme case of $w = 0.50\,\text{mm}$. This behavior can also be seen numerically in Table 7.2, where means and standard deviations of the presented distribution are shown in order to allow quantitative position resolution comparisons. An expected slight continuous decrease in z-position resolution with increased bin width can be observed, while the utilization of the $w = 0.05\,\text{mm}$ Grid Seeder configuration leads to the overall best longitudinal position resolution across all tested seed finder algorithms.
The transverse position resolution is as expected less prone to the chosen seed finding algorithm, where only a slight degradation can be observed for $w = 0.50\,\text{mm}$ in both Fig. 7.8 as well as Table 7.2.

[2] A $w = 0.50\,\text{mm}$ configuration corresponds to only $N_{\text{bins}} = 1000$ density bins along the beam axis in case of the non-adaptive Grid Seeder with a z-coverage of $[-250\,\text{mm}, 250\,\text{mm}]$.

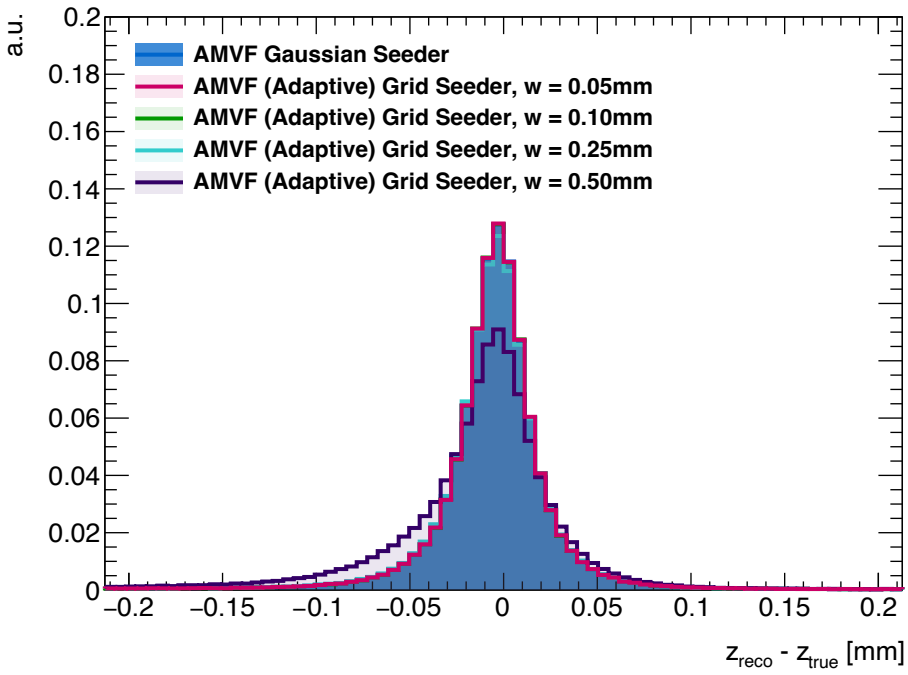


Figure 7.9: Resolution of z-positions of truth-matched vertices reconstructed by the ACTS AMVF deploying the Gaussian Seeder as well as different (Adaptive) Grid Seeder configurations. The truth-matching requires at least 50% of truth particles associated to the track content of the reconstructed vertex to have originated from a single true vertex.

	Seed Finder	Mean [mm]	Standard deviation [mm]
x-position	Gaussian Seeder	-1.4×10^{-5}	0.033
	Grid, $w = 0.05\,\text{mm}$	-3.1×10^{-5}	0.032
	Grid, $w = 0.10\,\text{mm}$	2.2×10^{-5}	0.032
	Grid, $w = 0.25\,\text{mm}$	2.6×10^{-5}	0.032
	Grid, $w = 0.50\,\text{mm}$	-4.0×10^{-5}	0.034
y-position	Gaussian Seeder	1.4×10^{-5}	0.033
	Grid, $w = 0.05\,\text{mm}$	-1.7×10^{-5}	0.032
	Grid, $w = 0.10\,\text{mm}$	-1.6×10^{-5}	0.032
	Grid, $w = 0.25\,\text{mm}$	-3.9×10^{-5}	0.032
	Grid, $w = 0.50\,\text{mm}$	-6.0×10^{-6}	0.034
z-position	Gaussian Seeder	-5.8×10^{-3}	0.043
	Grid, $w = 0.05\,\text{mm}$	-5.7×10^{-3}	0.042
	Grid, $w = 0.10\,\text{mm}$	-5.7×10^{-3}	0.043
	Grid, $w = 0.25\,\text{mm}$	-6.4×10^{-3}	0.044
	Grid, $w = 0.50\,\text{mm}$	-1.5×10^{-2}	0.056

Table 7.2: Mean and standard deviation values of the AMVF position resolution distributions for different vertex seed finding algorithms on 8000 $\mathrm{t\bar{t}}$ events in an average pile-up range from $\langle\mu\rangle = 20$ to 160.

The above presented vertex reconstruction efficiencies and position resolutions demonstrate that the newly developed (Adaptive) Grid Seeder offers a comparable, and therefore similarly excellent, physics performance as the Gaussian Seeder. An analysis of the (Adaptive) Grid Seeder's performance within the ATLAS reconstruction framework Athena will be presented in Section 7.3.4.

Seed Width Estimation Comparison

The estimation of the seed width is one of the outstanding properties of the Gaussian Seeder. A seed width estimation method for the (Adaptive) Grid Seeder based on the FWHM of the track density distribution was proposed in Section 5.4.3. Its performance in the context of a seed width constraint application in the AMVF will be compared to the Gaussian Seeder seed width estimation method in the following. Note that by default the seed width constraint is not applied in the AMVF algorithm, i.e. the above presented studies for evaluating the Grid Seeder's performance did *not* make use of any seed width constraint, whereas the constraint will be enabled in the AMVF for comparing the different estimation methods in the following.

Fig. 7.10 shows the distributions of estimated seed widths by the Gaussian Seeder as well as different (Adaptive) Grid Seeder configurations for the application within the AMVF algorithm on the same 8000 $t\bar{t}$ events as above. While the Gaussian Seeder predicts seed widths with an average of $\langle\sigma_z^2\rangle = 0.010\,\text{mm}^2$, the estimations obtained by the different Grid Seeder configurations are slightly larger with an expected trend towards higher values for coarser bin widths, ranging from $\langle\sigma_z^2\rangle = 0.012\,\text{mm}^2$ for the $w = 0.025\,\text{mm}$ configuration to $\langle\sigma_z^2\rangle = 0.019\,\text{mm}^2$ for $w = 0.100\,\text{mm}$.

A meaningful statement about the quality of the estimated seed width can again only be made by comparing the final reconstruction efficiencies of the AMVF algorithm utilizing the different seed finder algorithms while simultaneously applying the seed width constraint in the multi-vertex fit. The average number of reconstructed primary vertices using the AMVF algorithms with enabled seed width constraint and different vertex seed finding algorithms can be seen in Fig. 7.11. All AMVF (Adaptive) Grid Seeder configurations show physics performances very comparable to the one employing the Gaussian Seeder, while higher bin granularities result in slightly enhanced performances, as expected.

The resolutions of reconstructed and fitted vertex positions with the application of a seed width constraint have been studied and show a very similar behavior to the ones presented in Fig. 7.8 and Fig. 7.9 and are thus not explicitly shown again.

The seed width estimation technique developed for the (Adaptive) Grid Seeder algorithm is able to provide seed width values of a comparable quality compared to the Gaussian Seeder algorithm, making the (Adaptive) Grid Seeder a well-suited and equally performant alternative for primary vertex seed finding in terms of its physics performance. The CPU performances of the seed finding algorithms will be discussed in the following section.

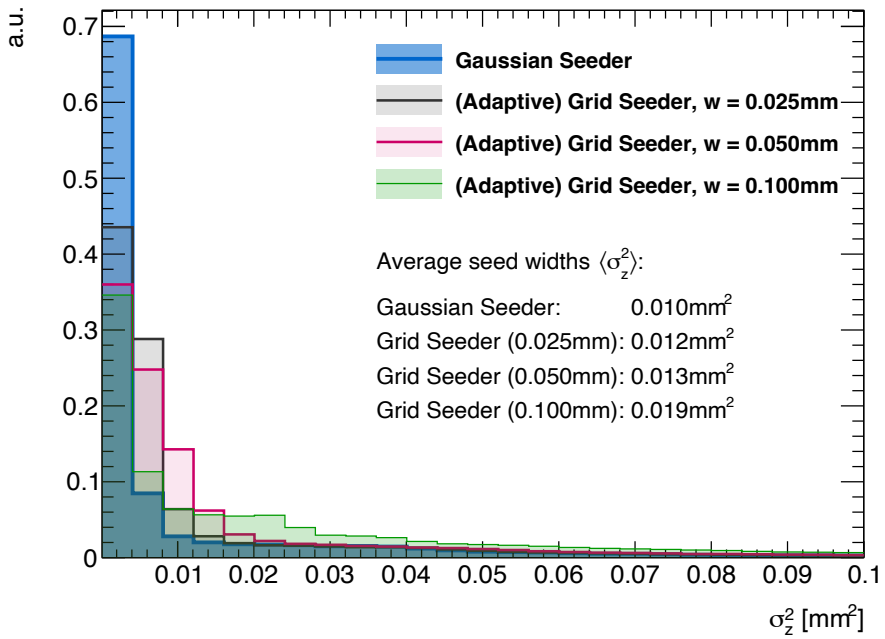


Figure 7.10: Distributions of estimated seed widths by the Gaussian Seeder and various (Adaptive) Grid Seeder configurations for the application within the AMVF algorithm on 8000 $t\bar{t}$ events in an average pile-up range from $\langle\mu\rangle = 20$ to 160.

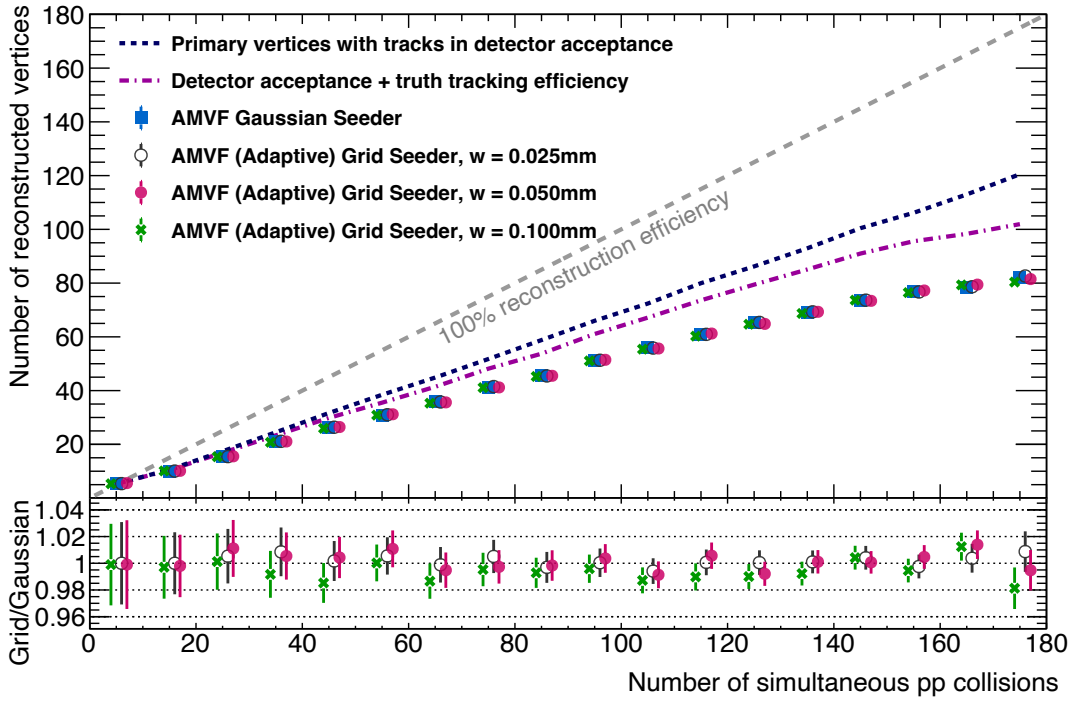


Figure 7.11: Average number of primary vertices reconstructed per event by the ACTS AMVF algorithm with enabled seed width constraint, employing the Gaussian Seeder as well as different (Adaptive) Grid Seeder configurations as a function of the true number of simultaneous *pp* collisions. A small per-bin offset for the different entries is applied, in order to improve the visibility of overlapping entries. For reference, a hypothetical 100% efficient reconstruction is indicated by a grey dashed line, while the other two dashed lines represent experimental reconstruction limits due to detector acceptance and tracking efficiency effects.

7.2.2 CPU Performance of the (Adaptive) Grid Seeder

The high granularity configurations of the (Adaptive) Grid Seeder show the same excellent physics performance as their Gaussian Seeder counterpart across a wide range of different pile-up scenarios. In the presence of a large number of simultaneous *pp* collisions ($\mu \gtrsim 60$), the Gaussian Seeder was found to show major, irreducible CPU performance issues in the past. The newly developed (Adaptive) Grid Seeder algorithm was specifically developed to overcome these severe limitations in high pile-up environments in order to prepare for the upcoming HL-LHC era and is therefore expected to outperform the Gaussian Seeder in these computationally challenging conditions. The CPU performance of the (Adaptive) Grid Seeder will be evaluated and compared to the Gaussian Seeder in the following.
As the algorithmic behavior and therefore also the expected CPU performance of the non-adaptive and adaptive versions of the Grid Seeder slightly differs, both variants will be studied separately. First, the non-adaptive Grid Seeder is employed as the seed finding method in the context of a full primary vertex reconstruction chain performed by the AMVF algorithm on the same 8000 $t\bar{t}$ events used above for the evaluation of the physics performance.
The average vertex seed finding CPU time required per event in the context of the AMVF primary vertex reconstruction is shown in Fig. 7.12 for the Gaussian Seeder as well as different Grid Seeder configurations. While for small numbers of simultaneous *pp* collisions the full primary vertex reconstruction spends well below 5 ms of CPU time on average per event on the vertex seed finding for both seed finding algorithm, a huge and very significant difference in primary vertex seed finding CPU time between the two algorithms starts to be apparent for larger pile-up values. The irreducible number of track density calculations in the case of the Gaussian Seeder, scaling more than quadratically with the number of tracks per event as reported in Section 5.4.2, leads to extremely large vertex seed finding times of up to 120 ms on average per event for $\mu \approx 180$. The sophisticated design of the Grid Seeder, on the other hand, requires only $N_{\text{bins}}^{\text{trk}}$ (i.e. a fixed number, usually around $10 - 200$ depending on the chosen configuration, see Appendix A.6) track density calculation per track and event and thus offers outstanding CPU performances also in high pile-up environments. Even the high granularity (and therefore more CPU intensive) versions of the Grid Seeder require only about 6 ms or less of average seed finding CPU time per event for $\mu \approx 180$, a speed-up of a factor of more than 20 compared to the Gaussian Seeder. The $w = 0.10$ mm configuration of the Grid Seeder, a configuration that was still able to show very similar physics performance compared to the Gaussian Seeder, can even reach seed finding CPU times that are more than 40 times faster in high pile-up conditions than the ones of the Gaussian Seeder algorithm.
The impact of the reduced vertex seed finding time on the overall primary vertex reconstruction time can be seen in Fig. 7.13, where the total average CPU time needed per event by the AMVF algorithm is shown for the utilization of the Gaussian Seeder as well as different Grid Seeder configurations.

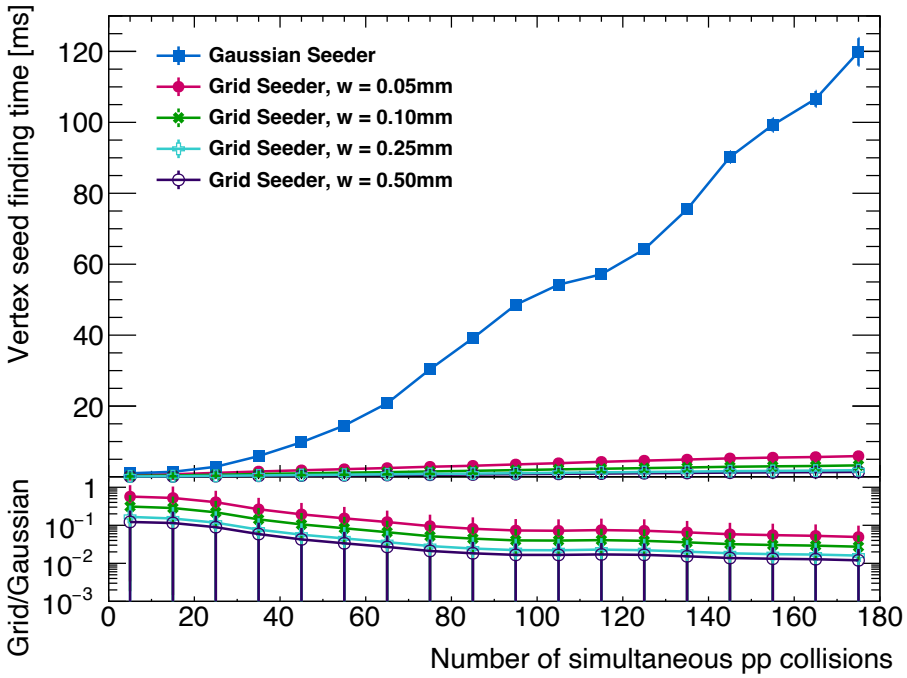


Figure 7.12: Average primary vertex seed finding CPU time required per event by the Gaussian Seeder and different Grid Seeder configurations in the context of the AMVF algorithm on 8000 $t\bar{t}$ events.

A significant improvement in total primary vertexing CPU time for high pile-up scenarios can be observed when utilizing the Grid Seeder within the AMVF algorithm, reducing the total required reconstruction time to about 60% of the one needed by the AMVF algorithm using the Gaussian Seeder. Although the coarse granularity Grid Seeder versions with $w = 0.25\,\text{mm}$ and $w = 0.50\,\text{mm}$ showed the fastest average seed finding CPU times in Fig. 7.12, their CPU time improvement impact on the overall primary vertex reconstruction time is diminished by a possible lower vertex seed quality: the very coarse bin width can occasionally lead to a vertex seed of bad quality, which can in turn result in an additionally required multi-vertex fit after the bad vertex estimate had been removed again from the list of fitted vertices, causing additional CPU overhead and therefore a degraded CPU time. This behavior can be seen in particular for the $w = 0.50\,\text{mm}$ Grid Seeder configuration in low pile-up conditions, where a significant CPU performance degradation is observed.
The overall best AMVF CPU performance across the full range of pile-up scenarios is seen for the utilization of the $w = 0.10\,\text{mm}$ Grid Seeder configuration, which can be regarded as the sweet spot bin width for guaranteeing high quality vertex seeds while keeping the amount of required track density calculations at a minimal level.
The $w = 0.05\,\text{mm}$ and $w = 0.10\,\text{mm}$ Grid Seeder configurations do therefore not only provide similarly excellent physics performance compared to Gaussian Seeder, but simultaneously outperform its CPU performance significantly in all studied pile-up conditions, making it a perfect choice for vertex seed finding for HL-LHC and beyond.

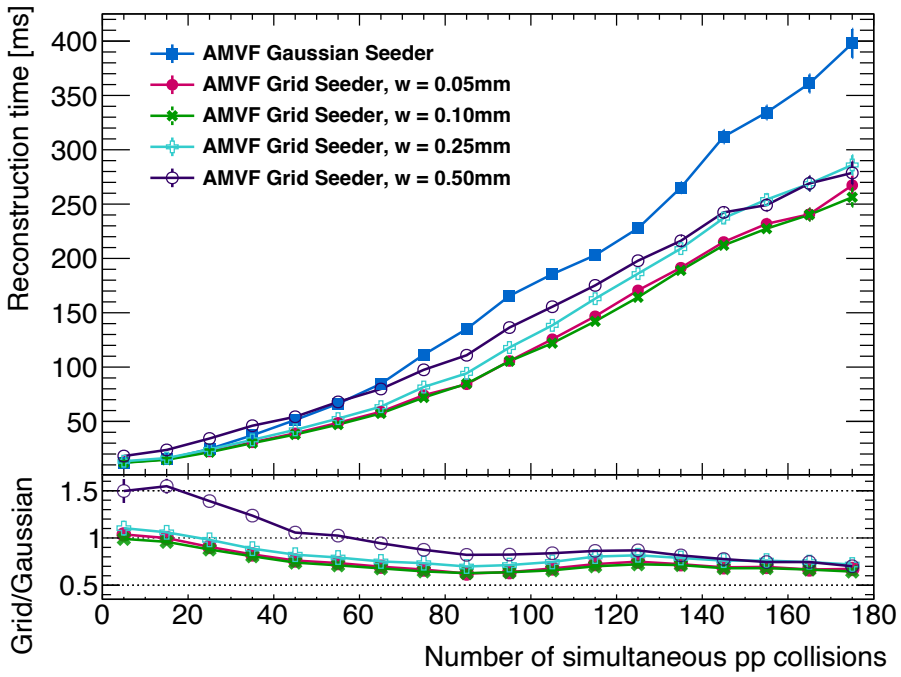


Figure 7.13: Average primary vertex reconstruction times needed per event as a function of the number of simultaneous *pp* collisions per bunch crossing for the ACTS AMVF algorithm employing the Gaussian Seeder as well as different Grid Seeder configurations.

CPU Performance of the Adaptive Grid Seeder and Memory Consumption Comparisons

The more sophisticated design of the Adaptive Grid Seeder allows for an optimized memory management, while additionally not requiring any hard z-boundary cuts anymore, comes however with an induced slight CPU performance overhead that can be seen in Fig. 7.14. Here, the total average primary vertex reconstruction CPU times needed per event by the AMVF algorithm are shown for the $w = 0.05\,\text{mm}$ and $w = 0.10\,\text{mm}$ Grid Seeder configurations, comparing the adaptive and non-adaptive versions. The utilization of the adaptive $w = 0.10\,\text{mm}$ version leads to an AMVF CPU time that is about 5% larger than the one when utilizing its non-adaptive counterpart. The relative increase in CPU time when replacing the non-adaptive Grid Seeder with the adaptive one is even more apparent for the $w = 0.05\,\text{mm}$ configuration.

In addition to the above presented CPU performance studies, the memory consumption of the full AMVF algorithm utilizing the Gaussian Seeder, the non-adaptive as well as the adaptive Grid Seeder algorithm was studied. As expected for the relatively low number of required track density bins along the z-axis for the $w = 0.05\,\text{mm}$ and $w = 0.10\,\text{mm}$ configurations, no significant differences in required memory could be observed between the utilizations of the Gaussian Seeder, non-adaptive Grid Seeder as well as the Adaptive Grid Seeder algorithm, as the vertex seed finders seemingly play only a minor role in the total memory consumption of the AMVF algorithm.
Unless a significantly higher number of bins along the z-axis is required (with $N_{\text{bins}} \gg$

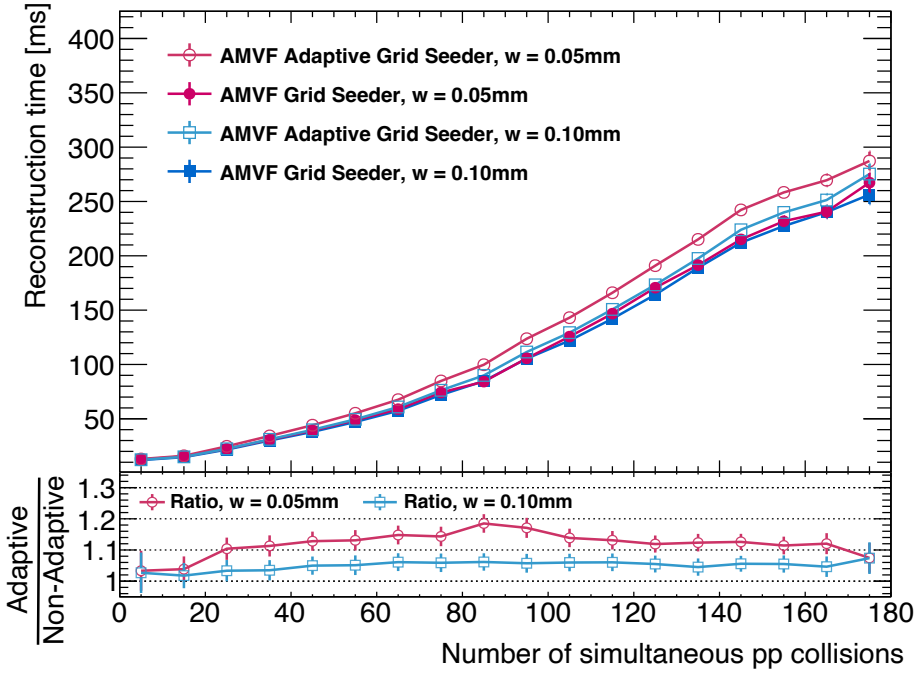


Figure 7.14: Average primary vertex reconstruction times needed per event as a function of the number of simultaneous *pp* collisions per bunch crossing for the ACTS AMVF algorithm employing different versions of the adaptive and non-adaptive Grid Seeder algorithm.

20 000 or more) and differences in required memory will therefore become more apparent, the choice between the application of the non-adaptive and adaptive Grid Seeder will mostly depend on whether flexible, dynamically growing z-boundaries are desired or not. In case of a collider experiment like ATLAS, where z-boundaries can easily be determined and the number of required track density bins along the z-axis will not exceed $N_{\text{bins}} \simeq 10\,000$, the non-adaptive Grid Seeder will be the method of choice as it provides the best possible CPU performances.

7.3 Application and Performance of the ACTS Vertexing in ATLAS

Aiming to provide a production-ready, modern and thread-safe primary vertex reconstruction software to ATLAS for the upcoming LHC Run 3 and beyond, the newly developed ACTS vertexing software was fully integrated, tested and validated in the ATLAS software framework Athena in the context of this thesis.

The technical work that has been done to integrate the ACTS vertexing software in Athena is discussed in Appendix A.7. Detailed physics validation and CPU performance studies of the ACTS AMVF, ACTS IVF as well as the Grid Seeder algorithm will be presented in the following.

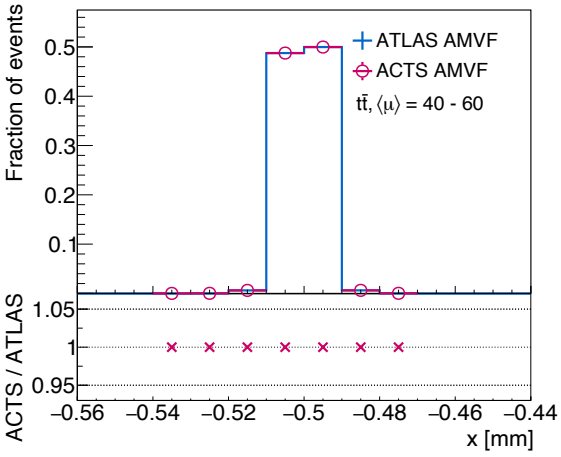
7.3.1 Single-Threaded Performance and Validation of the ACTS AMVF

In the following, single-threaded physics validation and CPU performance studies are presented for the ACTS AMVF application in Athena on 3000 $t\bar{t}$ events with a pile-up distribution between $\langle\mu\rangle = 40 - 60$. All studies presented in this section were additionally conducted on different MC samples such as $H \to \gamma\gamma$ and $Z \to \mu\mu$ as well as on ATLAS 2018 data. As identical behavior was observed on all tested samples, performance and validation studies are only presented for the application on $t\bar{t}$ events for the sake of conciseness.

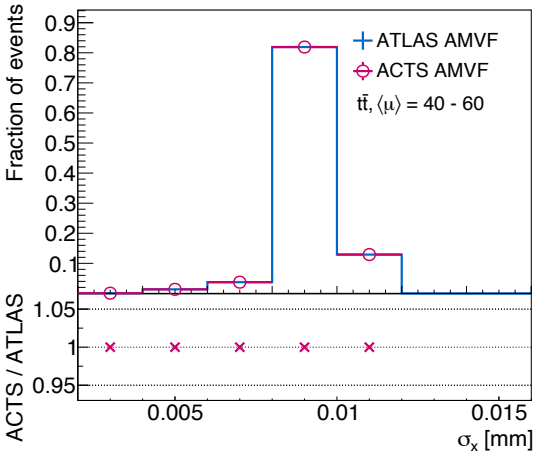
Physics Performance Validation

The physics performance of the ACTS AMVF was tested and validated against its original ATLAS implementation. Since a production usage of the ACTS vertexing software in ATLAS for LHC Run 3 and beyond is intended, rigorous testing and validation of its physics performance is needed. Hence, comparisons between the ACTS and ATLAS implementations for a variety of different quantities and variables for both all reconstructed primary vertices as well as the selected hard-scatter vertices are presented in the following.
Fig. 7.15 shows the position and position uncertainty distributions of all reconstructed primary vertices, comparing both the ACTS AMVF and the original ATLAS AMVF implementations deployed in the Athena software framework. A perfect agreement between the two illustrated vertex finder implementations can be seen. Note that the observed agreement does not imply numerically identical results. In fact, very small differences well below the illustrated comparison ratio scale are expected, as the two implementations make use of different underlying mathematical tools[3].
Perfect agreement can also be seen in Fig. 7.16, where the number of tracks associated with the reconstructed vertices, the track weight distribution as well as the truth-matched vertex type as defined in Section 5.2.5 are shown for all reconstructed primary vertices. Additionally, the hard-scatter reconstruction and selection efficiencies as well as the hard-scatter event classification as defined in Section 5.2.5 are shown with again perfect agreement between the two presented primary vertex finder implementations.
Also the selected HS vertices, which are usually the physics objects of interest, show perfect agreement between the ATLAS and ACTS AMVF implementations. Their z-positions, the number of associated tracks as well as their truth-matched vertex type and the reduced χ^2 distribution are shown in Fig. 7.17. For the sake of conciseness, several more HS variables are presented in Appendix A.8, again showing perfect agreement between the two implementations.

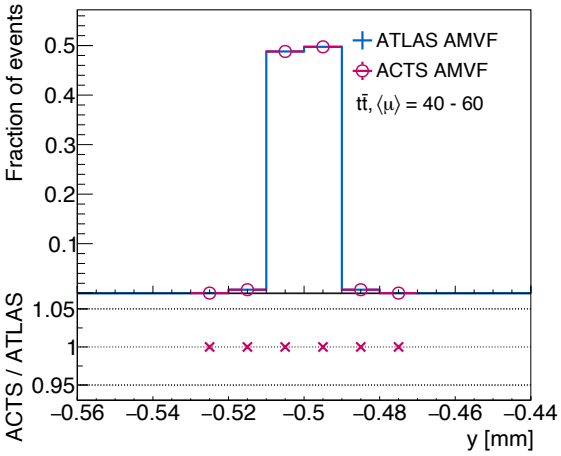
[3]While the ACTS vertex finder implementations make use of the `Acts::Propagator` tool for the extrapolation of track parameters, their ATLAS counterparts utilize the ATLAS `Trk::Extrapolator`, which can in general slightly differ in their functionalities.



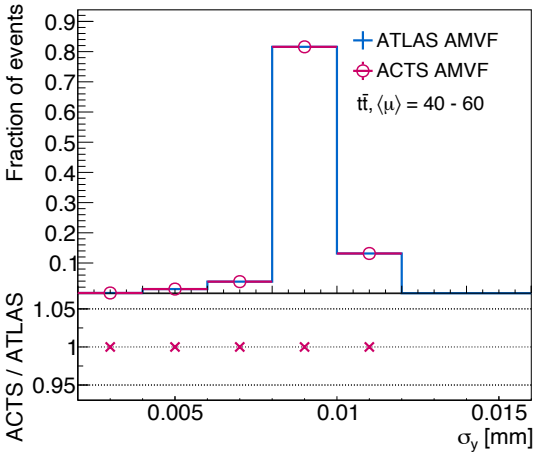
(a) All vertices – x-positions



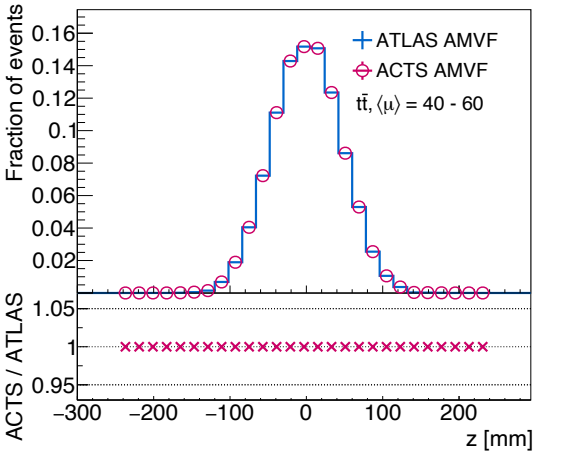
(b) All vertices – x-position errors



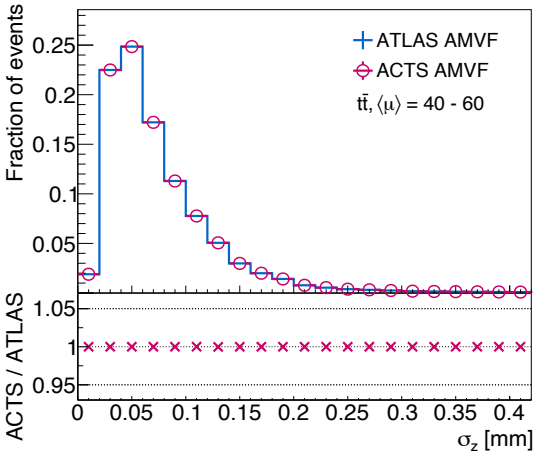
(c) All vertices – y-positions



(d) All vertices – y-position errors

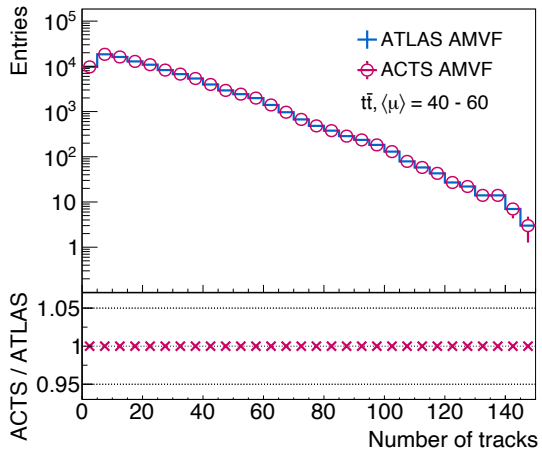


(e) All vertices – z-positions

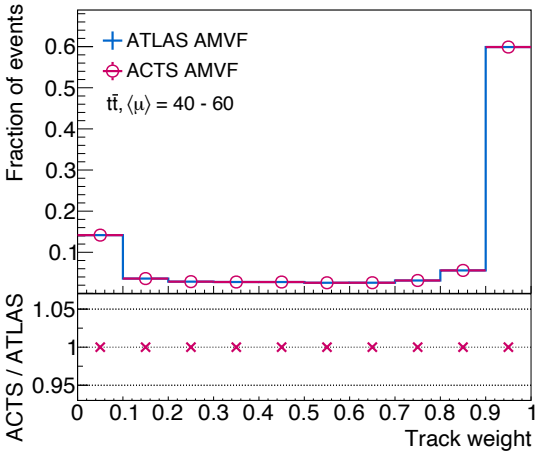


(f) All vertices – z-position errors

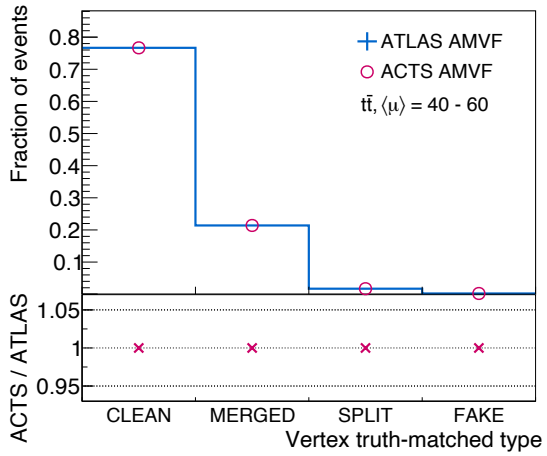
Figure 7.15: Position and position uncertainty distributions of all primary vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events in single-threaded execution mode, comparing the ACTS AMVF and ATLAS AMVF implementations.



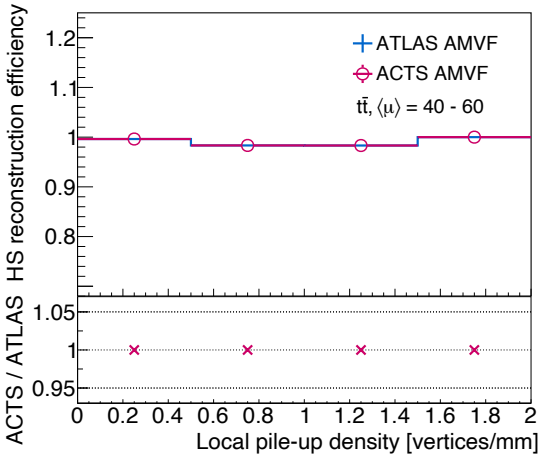
(a) All vertices – number of tracks



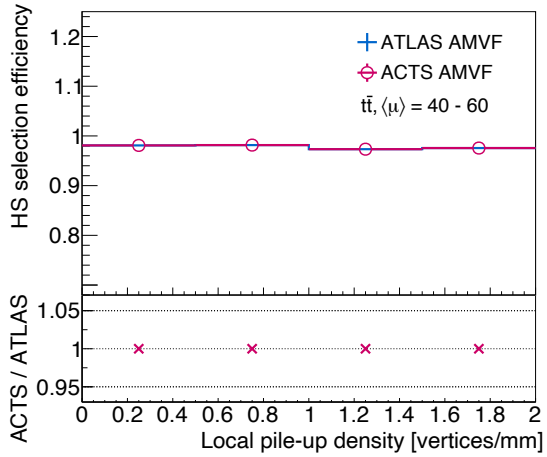
(b) All vertices – track weights



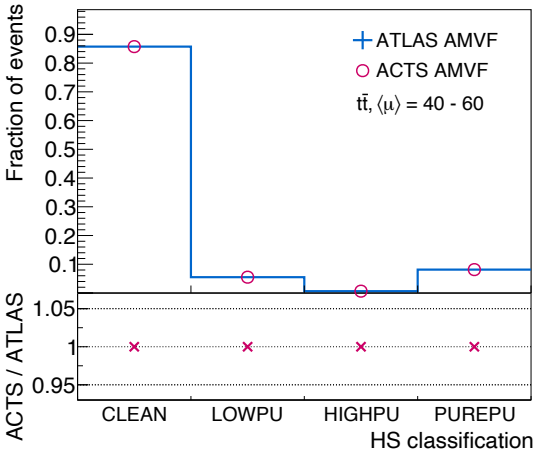
(c) All vertices – truth-matched type



(d) HS reconstruction efficiency

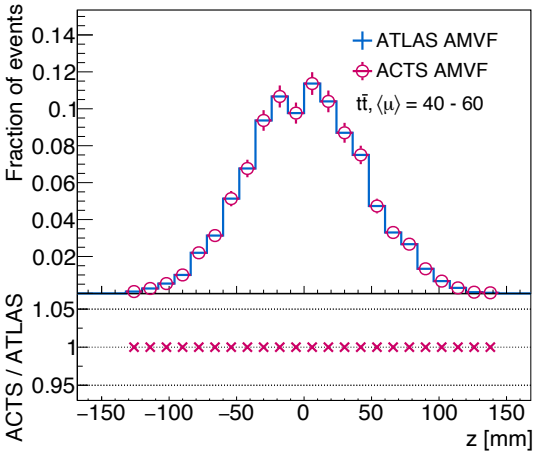


(e) HS selection efficiency

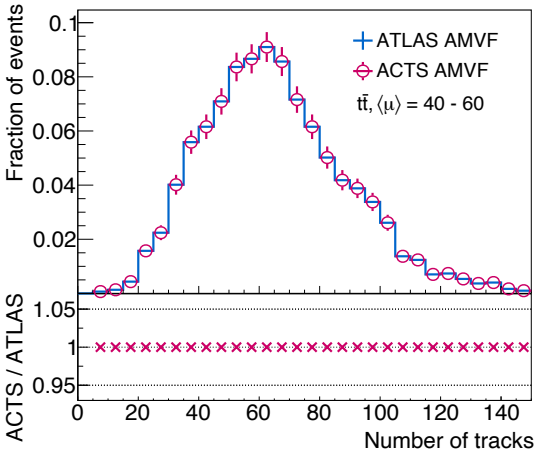


(f) HS classification

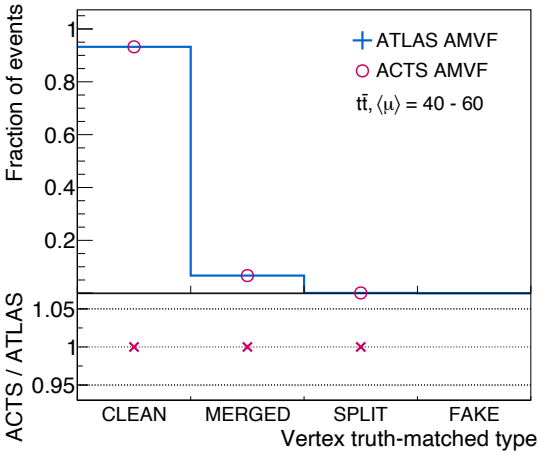
Figure 7.16: Comparison of the ACTS AMVF and ATLAS AMVF implementations, showing various different variables for all primary vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events in single-threaded execution mode.



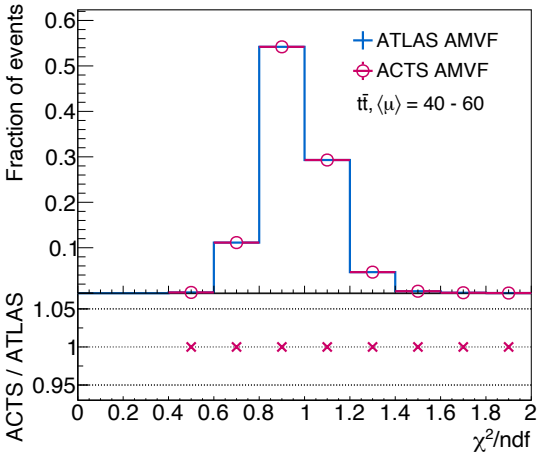
(a) Selected HS vertex – z-positions



(b) Selected HS vertex – number of tracks



(c) Selected HS vertex – truth-matched type



(d) Selected HS vertex – reduced χ^2

Figure 7.17: Comparison of the ACTS AMVF and ATLAS AMVF implementations, showing various different variables for the selected HS vertices in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events in single-threaded execution mode. The HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.

CPU Performance Comparison

Fig. 7.18 shows the total primary vertex reconstruction times needed per event for the ACTS AMVF and ATLAS AMVF algorithms employed in Athena. While the original ATLAS implementation needs up to 700 ms reconstruction time per event and averages 260 ms on the 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events, the ACTS implementation rarely exceeds 400 ms and exhibits an average of 125 ms reconstruction time, a CPU performance speed-up of more than a factor of two, as already seen in the CPU performance optimization studies discussed in Section 6.3.

This CPU performance improvement can also be seen in Fig. 7.19, where the average total reconstruction time needed per event as a function of the number of input tracks to the vertexing is shown for both AMVF implementations. The observed factor-two speed-up is almost constant across a wide range of track multiplicities and leads to a CPU time

reduction of more than 350 ms per event in dense environments. The ACTS AMVF hence shows a superb CPU performance, reducing the required reconstruction time by more than a factor of two, while simultaneously resulting in the exact same physics output as its original ATLAS implementation.

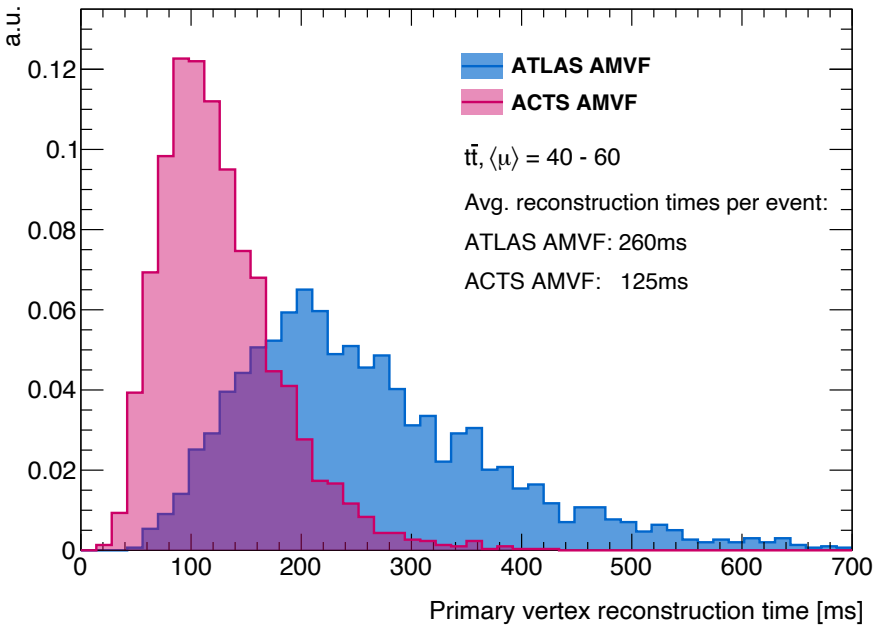


Figure 7.18: Total primary vertex reconstruction time per event for the ATLAS AMVF and ACTS AMVF algorithms on 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

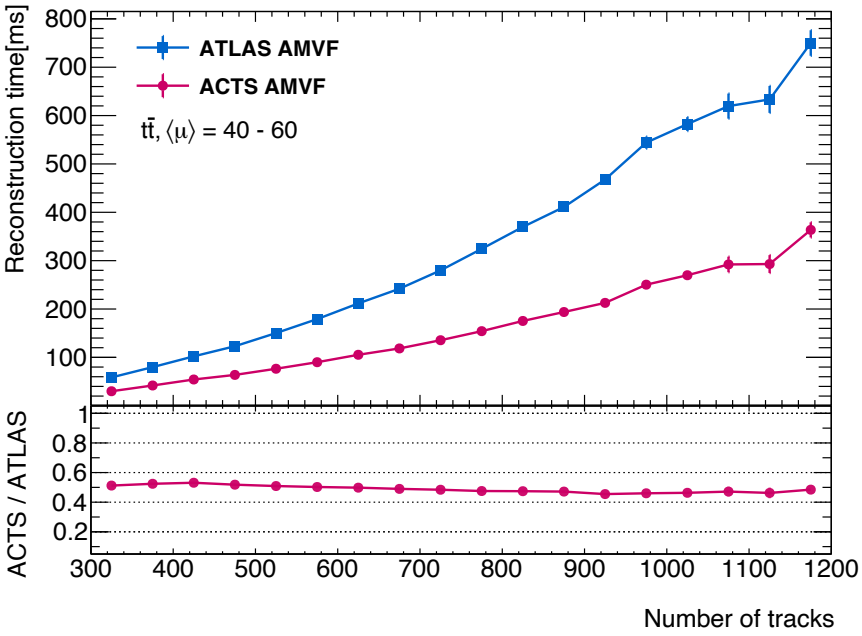


Figure 7.19: Average total primary vertex reconstruction times needed per event as a function of the number of tracks selected for primary vertexing, comparing the ATLAS AMVF and ACTS AMVF algorithms on 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

Furthermore, the relative contributions of the required EDM conversions, both the ATLAS-to-ACTS track conversion as well as the ACTS-to-ATLAS vertex conversion as discussed in Appendix A.7, to the above shown total primary vertex reconstruction times were studied by evaluating the CPU performance impact of both EDM conversions as well as the ACTS AMVF call itself, separately[4]. Fig. 7.20 shows these relative contributions to the total ACTS AMVF primary vertex reconstruction time as a function of the number of input tracks to the primary vertexing. The vast majority of the total ACTS AMVF CPU time in Athena is spent in the AMVF vertex reconstruction itself, while both EDM conversions only amount for less than 5% of the total CPU time for low track multiplicities. The relative impact of the EDM conversions further decreases to an almost negligible level with higher track multiplicities as the AMVF vertex reconstruction algorithm becomes increasingly more time consuming.

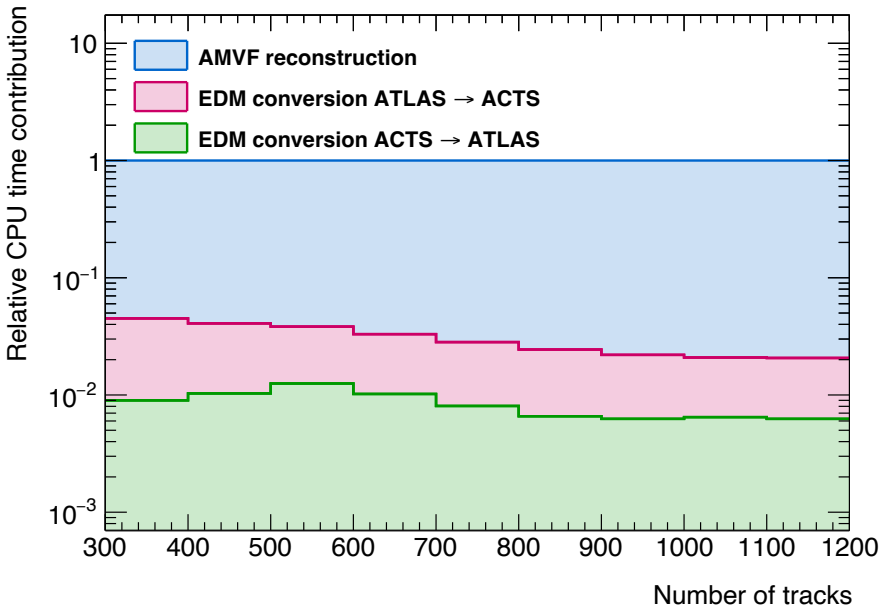


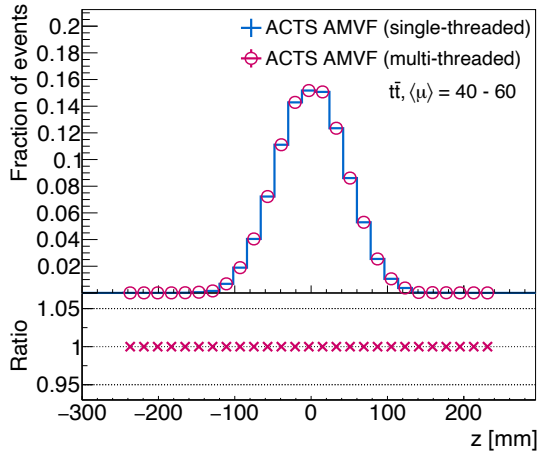
Figure 7.20: Relative contributions of the ACTS AMVF algorithm, ATLAS-to-ACTS EDM conversion and ACTS-to-ATLAS EDM conversion to the total ACTS AMVF primary vertex reconstruction time in Athena as a function of the number of tracks selected for primary vertexing. 100% correspond to the total ACTS AMVF CPU times presented in Fig. 7.19.

Across all shown track multiplicities, the ATLAS-to-ACTS EDM conversion has a greater CPU impact than its ACTS-to-ATLAS counterpart, simply because the number of track objects that need to be converted is generally much larger than the number of reconstructed vertex objects in the back-conversion. Note that once a full ACTS track reconstruction is available in ATLAS, the ATLAS-to-ACTS track EDM conversion will not be needed any longer as the resulting track objects will directly be `Acts::BoundTrackParameters` that can be used as input to the ACTS vertexing.

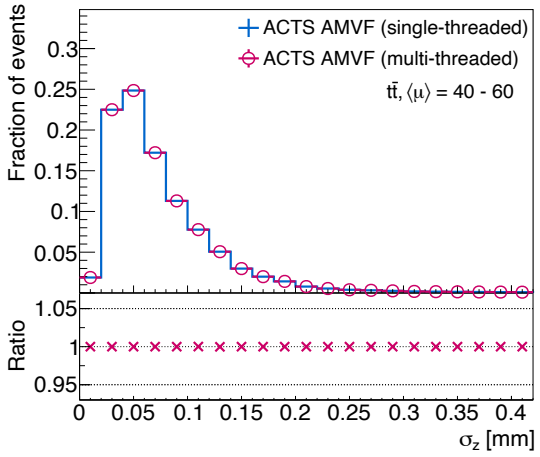
[4]For convenience, the sorting of the reconstructed vertices output list according to their HS compatibilities was chosen to be included in the ACTS-to-ATLAS vertex conversion CPU time contribution.

7.3.2 Multi-Threaded Performance and Validation of the ACTS AMVF

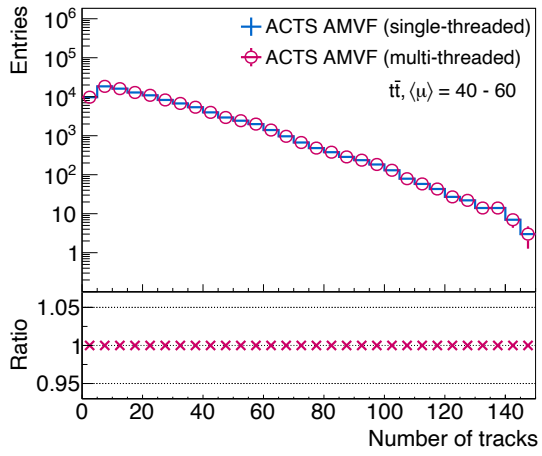
In addition to the validation of its single-threaded physics performance, the ACTS AMVF implementation also needs to be validated with respect to its performance in multi-threaded execution mode in Athena. Fig. 7.21 shows various different quantities for all reconstructed primary vertices in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events, comparing the ACTS AMVF application in Athena in single-threaded and multi-threaded execution mode. Several more variables tested on all reconstructed primary vertices are presented in Appendix A.9. Perfect agreement between the two presented algorithms can be seen in all studied quantities. The same agreement can be observed in Fig. 7.22, showing various different physics quantities for the selected HS vertices, comparing the single-threaded and multi-threaded ACTS AMVF vertex reconstruction. Several more HS variables are also presented in Appendix A.9 with



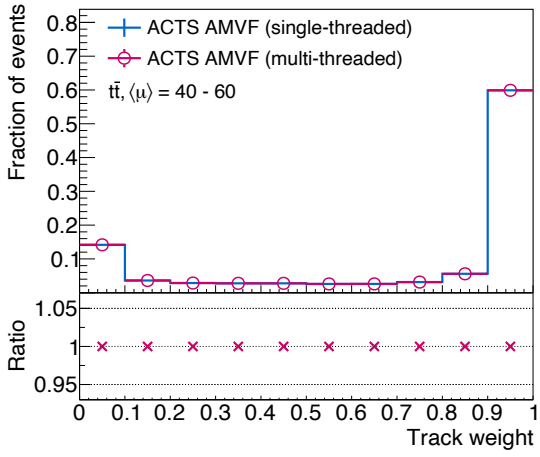
(a) All vertices – z-positions



(b) All vertices – z-position errors



(c) All vertices – number of tracks



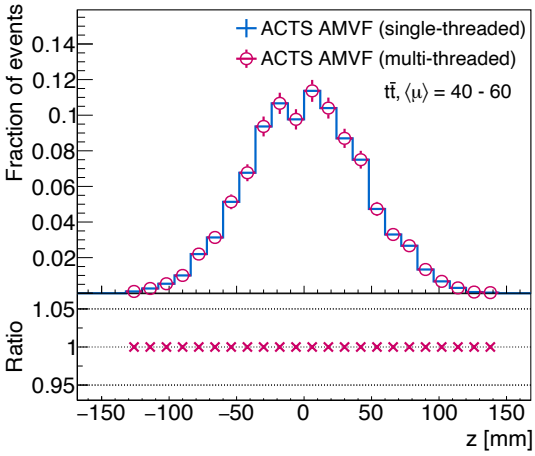
(d) All vertices – track weights

Figure 7.21: Comparison of the ACTS AMVF algorithm employed in Athena in single-threaded and multi-threaded execution mode, showing various different variables for all primary vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$. The multi-threaded execution was performed on eight concurrent threads.

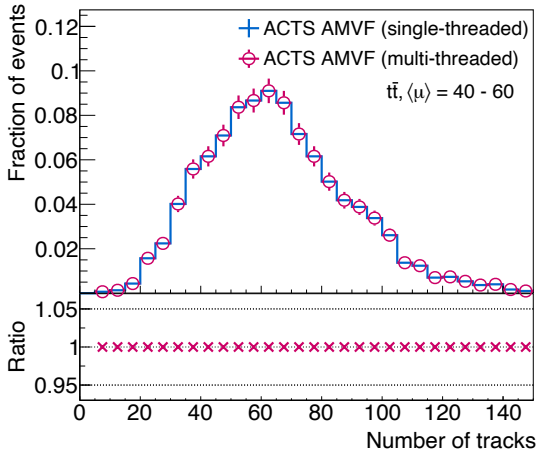
again perfect agreement.
The thread-safe ACTS AMVF implementation therefore allows for concurrent primary vertex reconstruction in ATLAS with the exact same physics results as the single-threaded ACTS as well as the original ATLAS AMVF implementations.

7.3.3 Single-Threaded Performance and Validation of the ACTS IVF

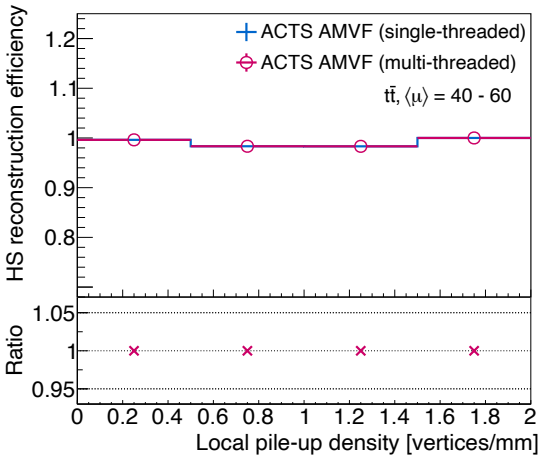
In the following, single-threaded physics validation and CPU performance studies are presented for the ACTS IVF application in ATLAS on 3000 $t\bar{t}$ events with a pile-up distribution between $\langle\mu\rangle = 40 - 60$. All studies presented in this section were additionally conducted on different MC samples such as $H \to \gamma\gamma$ and $Z \to \mu\mu$ as well as on ATLAS 2018 data. As identical behavior was observed on all tested samples, performance and validation



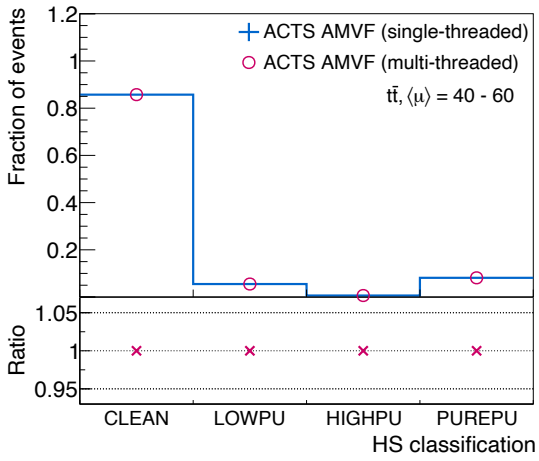
(a) Selected HS vertex – z-positions



(b) Selected HS vertex – number of tracks



(c) HS reconstruction efficiency



(d) HS classification

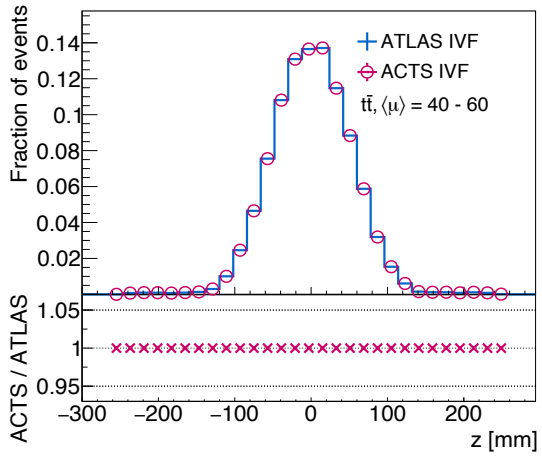
Figure 7.22: Comparison of the ACTS AMVF algorithm employed in Athena in single-threaded and multi-threaded execution mode, showing various different variables for the selected HS vertices reconstructed in 3000 $\langle\mu\rangle = 40-60$ $t\bar{t}$. The multi-threaded execution was performed on eight concurrent threads and the HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.

studies are only presented for the application on $t\bar{t}$ events for the sake of conciseness.

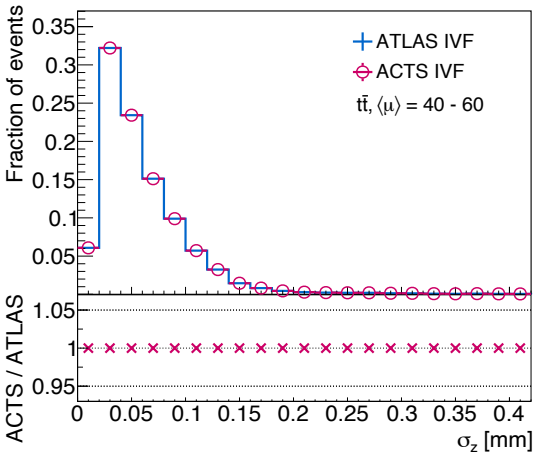
Physics Performance Validation

Fig. 7.23 shows various different quantities for all reconstructed primary vertices in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$, comparing the ACTS IVF and the original ATLAS IVF implementation. Several more variables tested on all reconstructed primary vertices are presented in Appendix A.10. Perfect agreement between the two presented algorithms can be seen in all studied quantities.

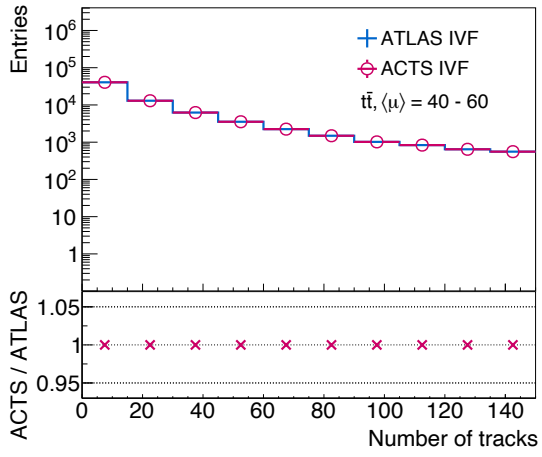
The same agreement can be observed in Fig. 7.24, showing various different physics quantities for the selected HS vertices, comparing the ACTS IVF implementation with its original ATLAS counterpart. Several more HS variables are also presented in Appendix A.10, showing again perfect agreement.



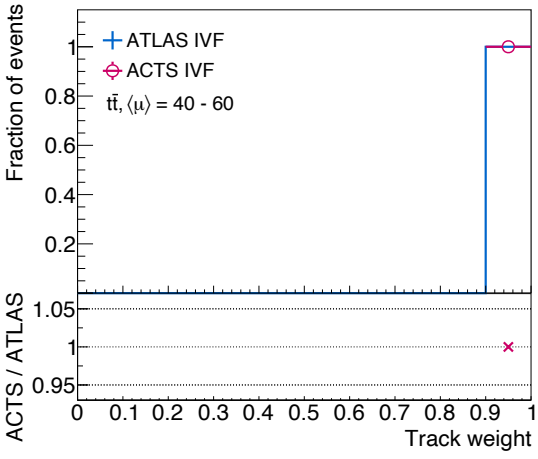
(a) All vertices – z-positions



(b) All vertices – z-position errors

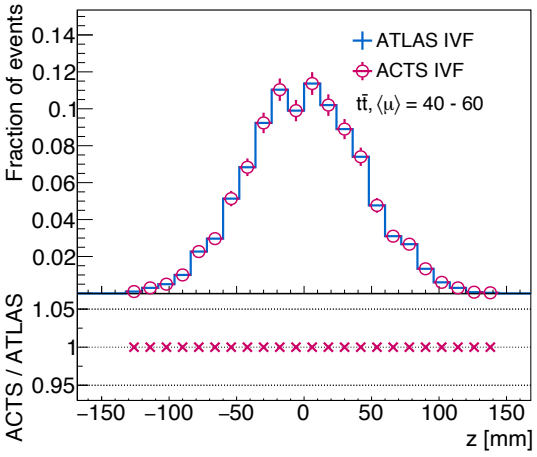


(c) All vertices – number of tracks

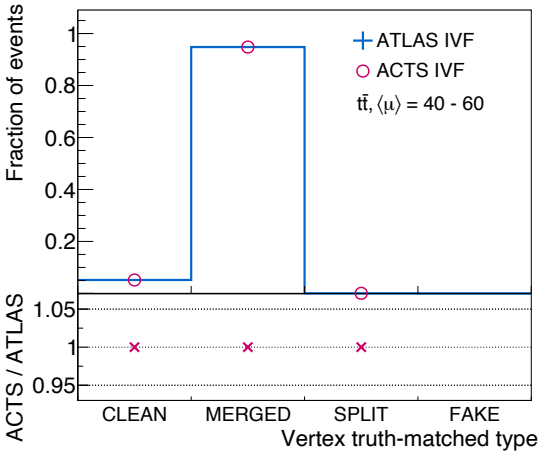


(d) All vertices – track weights

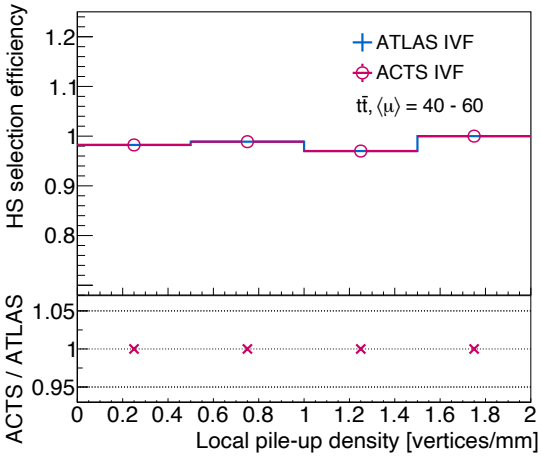
Figure 7.23: Comparison of the ACTS IVF and ATLAS IVF implementations, showing various different variables for all primary vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events in single-threaded execution mode.



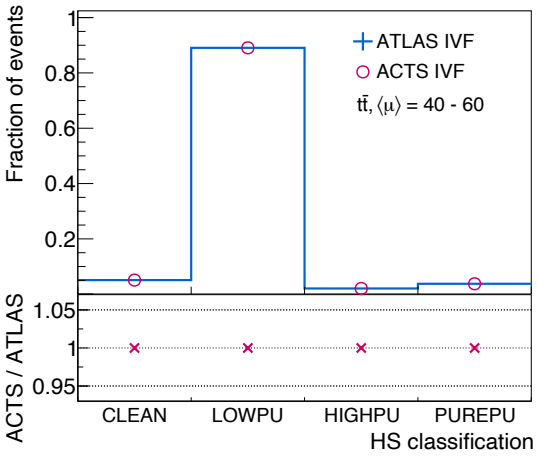
(a) Selected HS vertex – z-positions



(b) Selected HS vertex – truth-matched type



(c) HS selection efficiency



(d) HS classification

Figure 7.24: Comparison of the ACTS IVF and ATLAS IVF implementations, showing various different variables for the selected HS vertices in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events in single-threaded execution mode. The HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.

Also the multi-threaded application of the ACTS IVF in ATLAS showed again perfect agreement with its single-threaded counterpart and hence the original ATLAS IVF implementation. For the sake of conciseness, the corresponding validation studies are not explicitly shown in this thesis.

CPU Performance Comparison

Fig. 7.25 shows the total primary vertex reconstruction times needed per event for the ACTS IVF and ATLAS IVF algorithms. Also the ACTS IVF shows a significant CPU performance improvement compared to its original ATLAS implementation. While the ATLAS IVF algorithm averaged 102 ms reconstruction time on the tested 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events, the optimized ACTS implementation shows an average of only 61 ms.

This CPU performance improvement can also be seen in Fig. 7.26, where the average total reconstruction time needed per event as a function of the number of input tracks to the vertexing is shown for both IVF implementations. While for low track multiplicities the average CPU time required by the ACTS implementation is reduced to about 80% of the one needed by the ATLAS IVF algorithm, a reduction to about 50% can be seen in more dense environments.

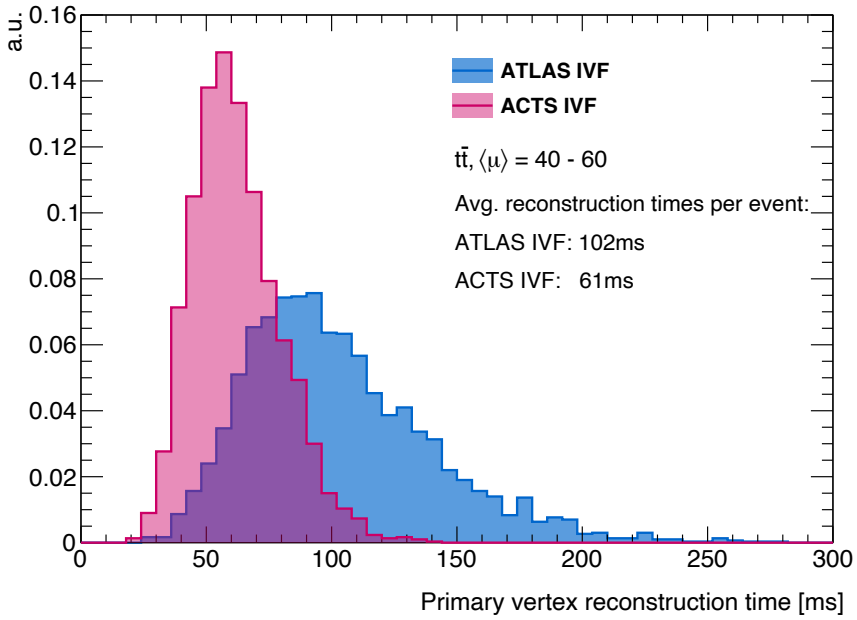


Figure 7.25: Total primary vertex reconstruction time per event for the ATLAS IVF and ACTS IVF algorithms on 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

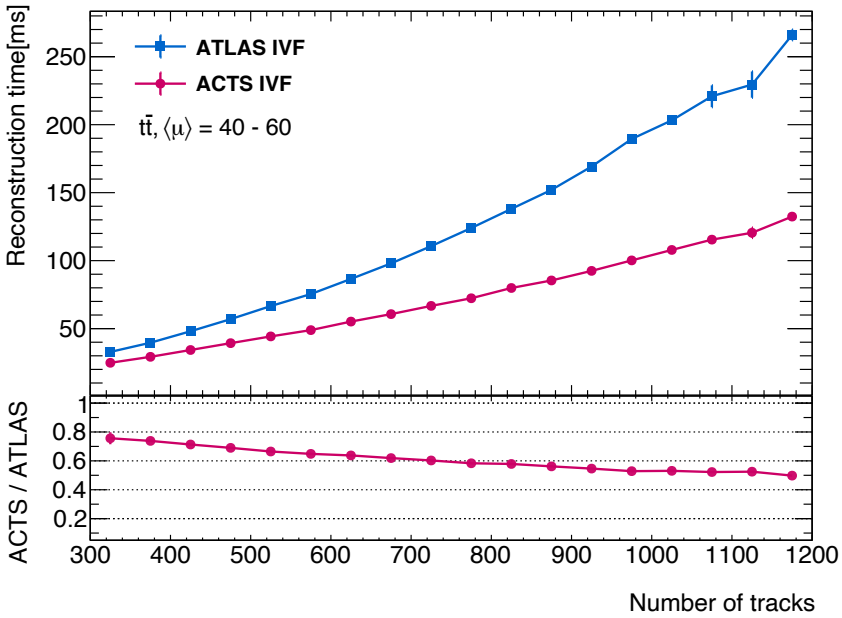


Figure 7.26: Average total primary vertex reconstruction times needed per event as a function of the number of tracks selected for primary vertexing, comparing the ATLAS IVF and ACTS IVF algorithms on 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

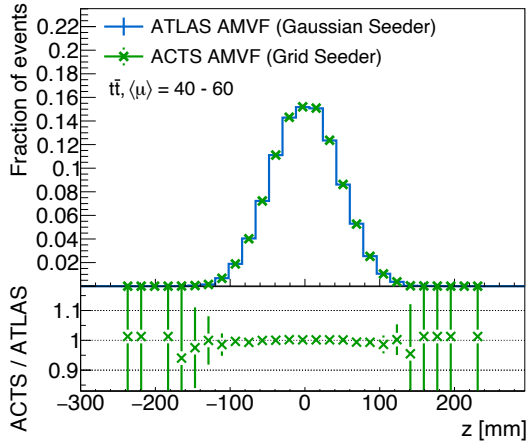
The observed speed-up is thus very significant, yet not as large as the one seen in the AMVF case. This is expected behavior, as all CPU performance improvement studies presented in Section 6.3 were specifically aiming to improve the computational performance of the AMVF implementation. CPU performance speed-ups in the IVF implementation, on the other hand, exclusively stem from optimization strategies that have been developed for the AMVF and could subsequently be adopted throughout the entire ACTS vertexing suite, such as the use of efficient data structures.

7.3.4 Performance of the ACTS Grid Seeder in ATLAS

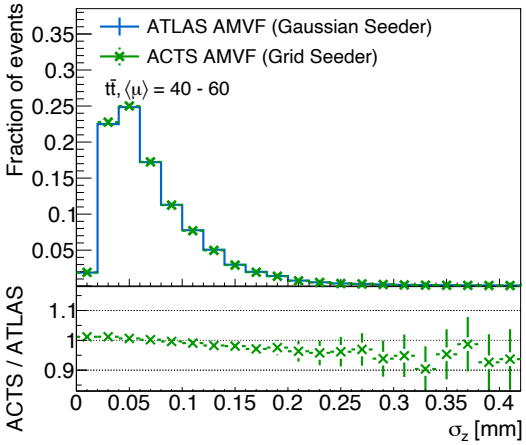
The newly developed Grid Seeder algorithm showed excellent physics and CPU performances in its application within the lightweight ACTS-Examples framework. An in-depth analysis in the context of a full primary vertex reconstruction chain in the ATLAS software framework will be presented in the following.
For this purpose, the ACTS AMVF algorithm is deployed with a $w = 0.05\,\text{mm}$ configuration of the Grid Seeder (i.e. $z_{\text{max}} = 250\,\text{mm}$ and $N_{\text{bins}} = 10\,000$), a configuration that led to the best physics results in the ACTS-Examples application while simultaneously exhibiting one of the fastest CPU times. Similar to the above presented validation studies for the AMVF and IVF implementations, the studies presented in the following are based on 3000 $\text{t}\bar{\text{t}}$ events with a pile-up distribution between $\langle\mu\rangle = 40-60$. All studies were also conducted on different MC samples such as $H \to \gamma\gamma$ and $Z \to \mu\mu$ as well as on ATLAS 2018 data. As identical behavior was observed on all tested samples, performance and validation studies are only presented for the application on $\text{t}\bar{\text{t}}$ events for the sake of conciseness.

Physics Performance

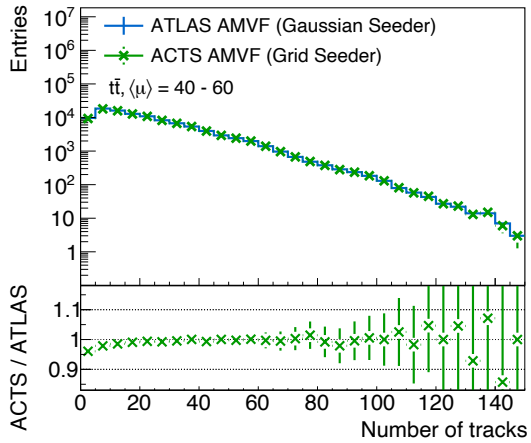
Fig. 7.27 shows the z-positions and z-position uncertainties, the number of tracks and track weights as well as other important physics quantities for all reconstructed primary vertices, comparing the ATLAS AMVF using the Gaussian Seeder as its vertex seed finding algorithm and the ACTS AMVF algorithm employing the $w = 0.05\,\text{mm}$ Grid Seeder. As expected, the resulting distributions slightly differ as different seed finding methods are being used, but generally show very similar behavior and physics performance.
The same can be seen in Fig. 7.28 for the selected hard-scatter vertices: the resulting distributions of the reconstructed z-positions as well as the number of tracks at vertex are almost identical, while even a slight improvement in physics performance can be observed for the Grid Seeder utilization when comparing the resulting distributions of the hard-scatter classification, indicating that the resulting hard-scatter vertices are of higher quality when utilizing the Grid Seeder, as the contamination of pile-up vertices is slightly reduced. In terms of hard-scatter reconstruction and selection efficiency, the new Grid Seeder shows the same excellent physics performance as its Gaussian counterpart.
More physics performance comparisons of various different quantities for both all reconstructed as well as the selected hard-scatter vertices can be found in Appendix A.11.



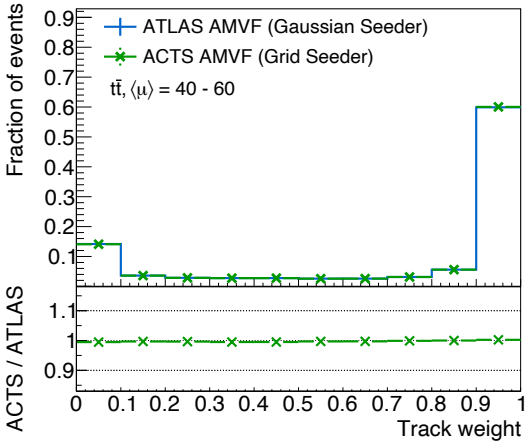
(a) All vertices – z-positions



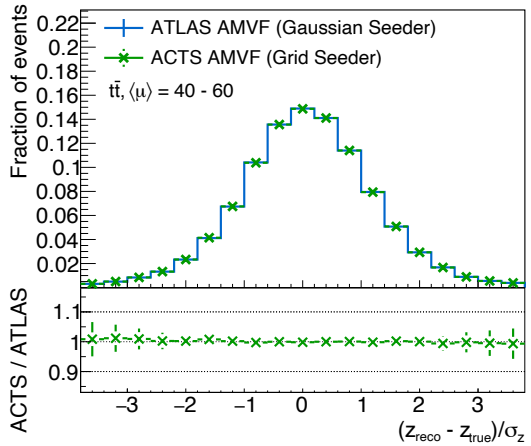
(b) All vertices – z-position errors



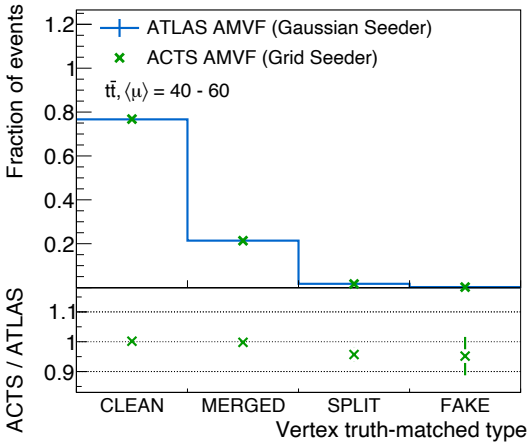
(c) All vertices – number of tracks



(d) All vertices – track weights

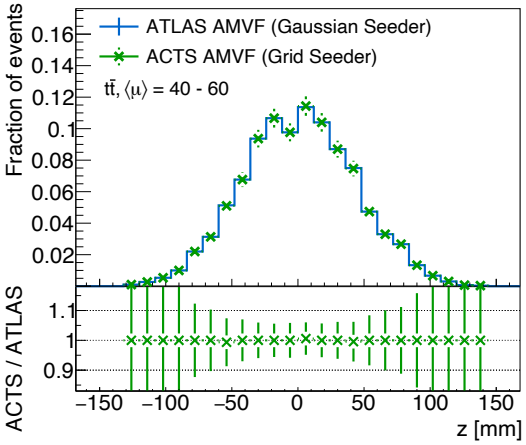


(e) All vertices – z-pull

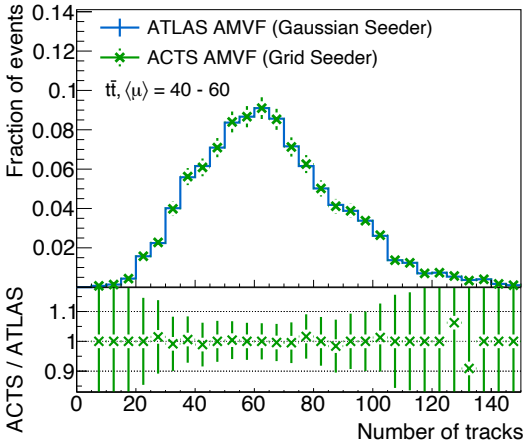


(f) All vertices – truth-matched type

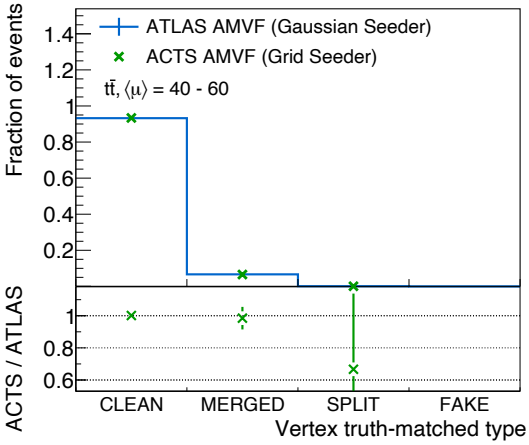
Figure 7.27: Comparison of the ATLAS AMVF algorithm utilizing the Gaussian Seeder and the ACTS AMVF algorithm utilizing a $w = 0.05\,\text{mm}$ Grid Seeder configuration, showing various different variables for all primary vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $\text{t}\bar{\text{t}}$ events in single-threaded execution mode.



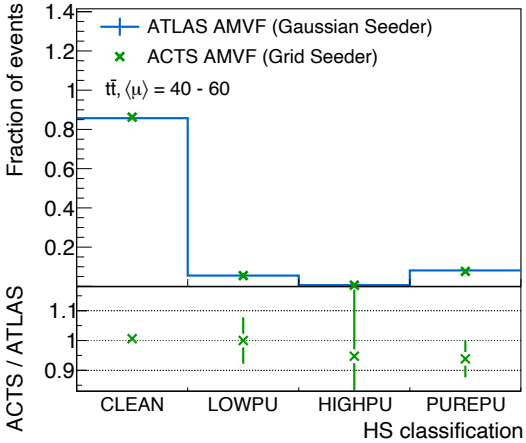
(a) Selected HS vertex – z-positions



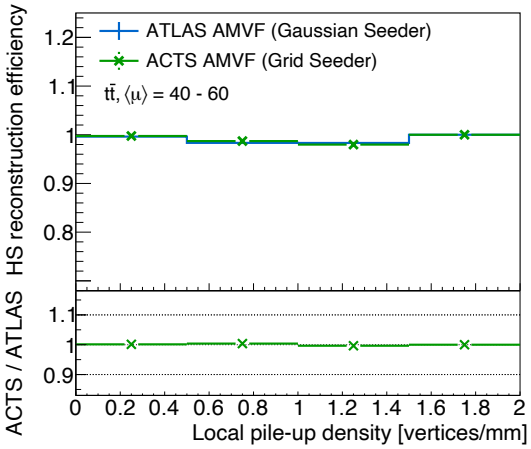
(b) Selected HS vertex – number of tracks



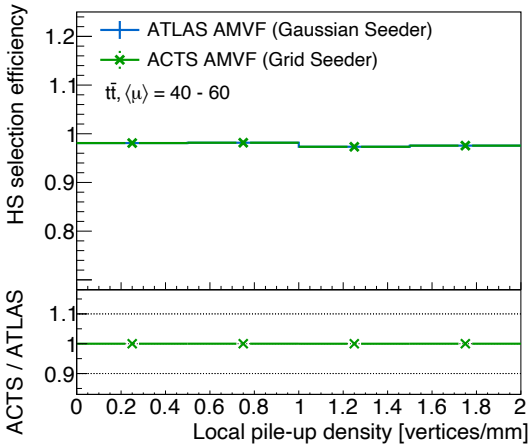
(c) Selected HS vertex – truth-matched type



(d) HS classification



(e) HS reconstruction efficiency



(f) HS selection efficiency

Figure 7.28: Comparison of the ATLAS AMVF algorithm utilizing the Gaussian Seeder and the ACTS AMVF algorithm utilizing a $w = 0.05\,\text{mm}$ Grid Seeder configuration, showing various different variables for the selected HS vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $\text{t}\bar{\text{t}}$ events in single-threaded execution mode. The HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.

CPU Performance Comparison

The Grid Seeder algorithm was developed in the course of this thesis to overcome the severe CPU performance limitations seen for the Gaussian Seeder method, specifically in dense, high pile-up environments. Fig. 7.29 shows the total primary vertex reconstruction times needed per event, comparing the ATLAS AMVF (Gaussian Seeder), the ACTS AMVF (Gaussian Seeder) as well as the ACTS AMVF (Grid Seeder) algorithms. While the performance-optimized ACTS AMVF (Gaussian Seeder) implementation could already reduce the total required CPU time per event from 260 ms to 125 ms, as discussed above in Section 7.3.1, the deployment of the Grid Seeder algorithm within the ACTS AMVF leads to an outstanding overall CPU performance with an average total primary vertex reconstruction time of less than 90 ms on 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.
The Grid Seeder's biggest strength becomes visible in Fig. 7.30, where the average total reconstruction time needed per event as a function of the number of input tracks to the vertexing is shown for the ATLAS AMVF (Gaussian Seeder), the ACTS AMVF (Gaussian Seeder) as well as the ACTS AMVF (Grid Seeder) algorithm. Especially in dense environments with large track multiplicities, i.e. generally high pile-up environments, the deployment of the Grid Seeder algorithm leads to CPU performance speed-ups of more than a factor of 3.5 compared to the Gaussian Seeder in the ATLAS AMVF, reducing the required CPU time from more than 750 ms per event to less than 200 ms. As the relative speed-up compared to the Gaussian Seeder deployment becomes larger with increasing track multiplicities in the presented $\langle\mu\rangle = 40 - 60$ environments, much greater relative speed-ups can be expected in the upcoming high luminosity environments in HL-LHC with $\langle\mu\rangle \approx 200$.

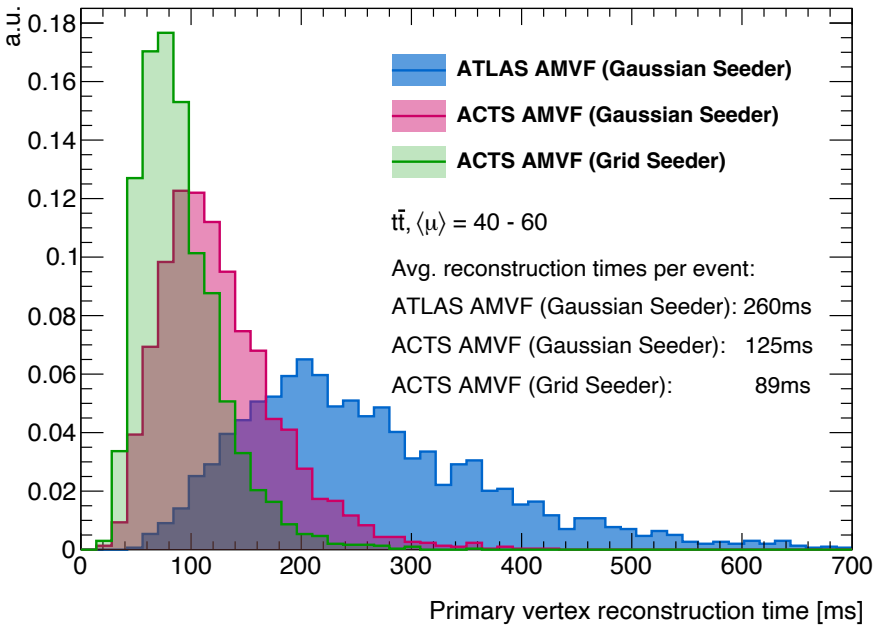


Figure 7.29: Total primary vertex reconstruction time per event for the ATLAS AMVF (Gaussian Seeder), ACTS AMVF (Gaussian Seeder) and ACTS AMVF (Grid Seeder, $w = 0.05$ mm) algorithms on 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

In addition to its excellent physics performance, that was shown to be comparable with or even slightly better in terms of the hard-scatter reconstruction quality than the Gaussian Seeder, the newly presented Grid Seeder clearly outperforms the CPU performance of the Gaussian Seeder in all tested environments with a particular strength in high pile-up conditions, making it a perfect choice for primary vertex seed finding in the upcoming future HL-LHC conditions.

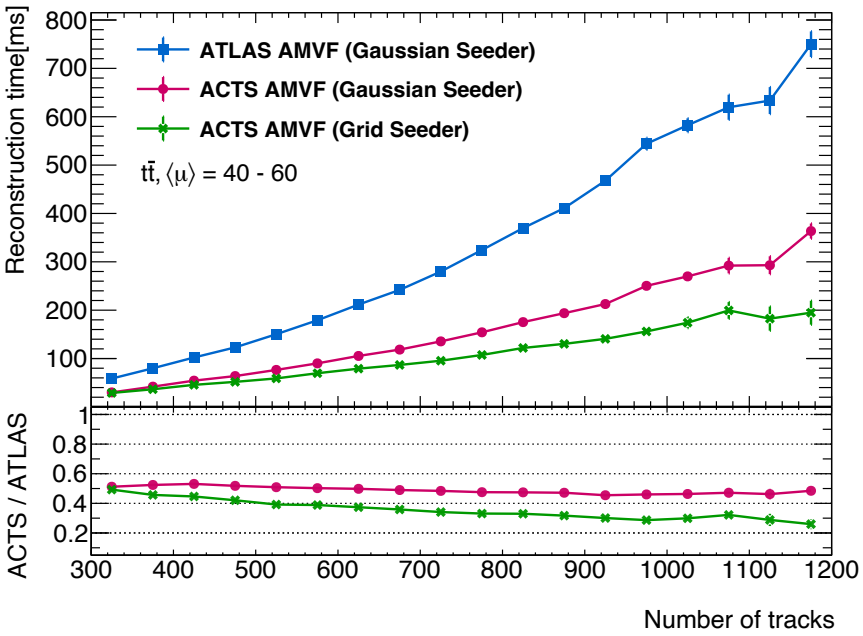


Figure 7.30: Average total primary vertex reconstruction times needed per event as a function of the number of tracks selected for primary vertexing, comparing the ATLAS AMVF (Gaussian Seeder), the ACTS AMVF (Gaussian Seeder) and the ACTS AMVF (Grid Seeder, $w = 0.05\,\text{mm}$) algorithms on 3000 $\langle\mu\rangle = 40 - 60$ $\text{t}\bar{\text{t}}$ events.

Due to its superb physics and CPU performance as well as its inherently thread-safe design, the above presented ACTS AMVF (Gaussian Seeder) vertexing software will be used as the default primary vertex reconstruction tool in ATLAS for LHC Run 3, as reported in Ref. [122]. It therefore marks the first production use of an ACTS software component in an LHC experiment.

Additionally, ATLAS is currently planning to deploy the presented Grid Seeder algorithm for vertex seed finding in the upcoming high pile-up environments for HL-LHC due to its excellent CPU and physics performance capabilities in these challenging conditions.

Furthermore, also several other HEP experiments, such as sPHENIX [123, 124], FASER [125] or the HL-LHC Beam Gas Vertex Monitor (BGV) [126] are already using or preparing to be using the ACTS vertexing software developed in the context of this thesis as part of their event reconstruction software.

Part III

Search for Flavor-Violating Supersymmetry with the ATLAS Experiment

Chapter 8

Common Analysis Concepts in Searches for New Physics

A careful and thorough analysis of the large data samples recorded and reconstructed by the ATLAS experiment is crucial in order to precisely study the properties of fundamental particles and their interactions or, as presented in the physics analysis in Chapter 9, to make conclusive statements about the existence of a new particle or the validity of a new theory beyond the Standard Model. Depending on the stated goal of the data analysis, i.e. for example the desired measurement of a certain quantity up to a specific level of precision or the search of new physics in a certain final state, the exact analysis details can of course be very different. Most of them, however, follow general concepts, or *analysis strategies*, and make use of common statistical methods that will be outlined in this chapter with particular focus on the concepts commonly used in searches for new physics.

8.1 Common Analysis Strategy

8.1.1 Signal Regions

A data analysis searching for new physics typically starts with the definition of a final state in which the new physics model will be tested and requires a precise understanding of the signatures that the physics processes or particles predicted by the theory to be tested will leave inside the particle detector. The goal of the analysis is then to analyze the recorded datasets for these specific signatures, or *signals*, while rejecting already known Standard Model *background* processes.

Typically, dedicated variables describing the kinematic properties of a collision event, such as the missing transverse momentum, the number of jets, angular distances between certain objects or more elaborated mass variables, are defined and evaluated for every single event. Each event can therefore be thought of as a point representation in a multidimensional phase space while events with similar kinematic properties will populate similar, distinct regions that can be used to separate background events from the signal events of interest in the specific analysis.

The goal of a so-called cut-and-count analysis is thus to apply selections cuts (also often just referred to as *cuts*) on kinematic variables to target a specific region of phase space where

as much signal as possible is expected, while simultaneously keeping the contamination of background processes at a minimum. Often, machine learning-based approaches such as deep neural networks or boosted decision trees are additionally employed to further separate signal from background events. The resulting region of phase space with an increased signal purity is referred to as *signal region* (SR).
A new theory to be tested often involves unknown model parameters, such as the masses of predicted particles, which can lead to very different kinematic properties of the signal events and thus to different signal topologies, depending on the choice of these parameters. It is therefore common to define not only one single SR but often multiple SRs are defined, each one targeting a dedicated region of phase space, where a signal enhancement for a specific signal topology is expected.

8.1.2 Background Estimation and Validation

Once a SR, or more general a set of SRs, is defined, a detailed understanding of how the Standard Model background processes behave in these regions must be acquired. The predicted behavior in the SRs can then be compared to the actual data collected by the ATLAS experiment, allowing to make conclusive statements about whether or not the observed data is likely to contain a certain signal process by means of statistical methods discussed in more detail in Section 8.2. The precise estimation of the background processes entering the defined SRs is thus a crucial step in a physics analysis and commonly used approaches as well as their validation will be briefly discussed in the following.
While many different background estimation techniques exist, most of them rely on Monte Carlo simulations of the physics processes of interest with detailed detector simulations as already discussed in Section 3.4. These Monte Carlo simulations can be used to predict the background contributions of different Standard Model processes in the SRs and to determine the dominant background processes, i.e. processes with the largest relative contributions in a SR, as well as the sub-dominant ones, i.e. the processes which contribute to the overall background only at a small or even negligible level.
Especially the dominant backgrounds processes and their behaviors in the SRs, i.e. their overall predicted rates as well as their shapes in the discriminating variables, need to be very well understood. However, SRs are often defined in rather extreme domains of the phase space in which the underlying theoretical concepts, and thus also the simulated background predictions, may not be accurately modeled any longer. In order to better estimate and *control* the dominant background processes contaminating the SRs, and therefore increase the confidence in these background predictions, a semi-data-driven background estimation approach using so-called *control regions* (CRs) is often employed [127].
CRs are designed to have a high purity in one specific type of background process and as little as possible signal contamination, while simultaneously being kinematically close, yet orthogonal, to the SRs. Two regions are said to be orthogonal to one another if they are statistically independent and thus no event can simultaneously satisfy both region selection requirements.
The MC predicted event yield of a certain background process p in its dedicated CR can

be compared and *fit* (see Section 8.2.2) to the actual ATLAS data in this region, allowing to account for MC normalization mismodeling by scaling the initial predictions to the actual observed levels using normalization factors μ_p that are computed during the fitting procedure. These background normalizations are then extrapolated to the SRs in order to retrieve so-called *normalized background predictions* and thus gain a more accurate understanding of the SM backgrounds of interest in the SRs. The normalized SR background prediction for the physics process p is now given by

$$N_p^{\mathrm{SR}} = \underbrace{\left(\frac{N_{p,\mathrm{data}}^{\mathrm{CR}}}{N_{p,\mathrm{MC}}^{\mathrm{CR}}}\right)}_{\mu_p} \times N_{p,\mathrm{MC}}^{\mathrm{SR}} = \mu_p \times N_{p,\mathrm{MC}}^{\mathrm{SR}} \tag{8.1}$$

$$= \underbrace{\left(\frac{N_{p,\mathrm{MC}}^{\mathrm{SR}}}{N_{p,\mathrm{MC}}^{\mathrm{CR}}}\right)}_{\mathrm{TF}_p} \times N_{p,\mathrm{data}}^{\mathrm{CR}} = \mathrm{TF}_p \times N_{p,\mathrm{data}}^{\mathrm{CR}}, \tag{8.2}$$

where $N_{p,\mathrm{data}}^{\mathrm{CR}}$ denotes the number of observed background events in the CR corrected for the contamination from processes other than p, $N_{p,\mathrm{MC}}^{\mathrm{CR}}$ and $N_{p,\mathrm{MC}}^{\mathrm{SR}}$ the initially predicted contributions from process p in the CR and SR, respectively, and TF_p the so-called *transfer factor*.

Although the mathematical fitting procedure internally uses Eq. (8.1) for evaluating N_p^{SR}, these normalized SR background predictions can be interpreted according to Eq. (8.2) as a product of a transfer factor TF_p and $N_{p,\mathrm{data}}^{\mathrm{CR}}$. A very important feature of the transfer factors is that, by virtue of using a ratio of MC estimates, systematic uncertainties on the MC predicted backgrounds can fully or partially cancel out during the extrapolation from the CR to the SR. The combination of the statistical uncertainty on the number of background events in the CR and the residual systematic uncertainties thus defines the overall uncertainty on the number of normalized background events in the SR [127].

When estimating the number of background events in the SR based on an extrapolation from the CR, the underlying assumption is made that this extrapolation is well modeled and will thus result in a reliable background estimation, as the CRs are populating domains of phase space that are close to the ones of the SRs. In order to validate the SR background estimation strategy, so-called *validation regions* (VRs) are defined. VRs are typically placed in the phase space region between the CRs and SRs that is extrapolated over, allowing to validate the normalization factors derived in the CRs in a phase space close to the SRs by comparing the predicted background yields with observed data. Fig. 8.1 schematically illustrates the concept of extrapolation in two discriminating variables between two CRs and two SRs with two intermediate VRs. All shown regions are designed to be statistically independent of one another and can additionally be binned in the their defining variables which is not depicted in the illustration [127].

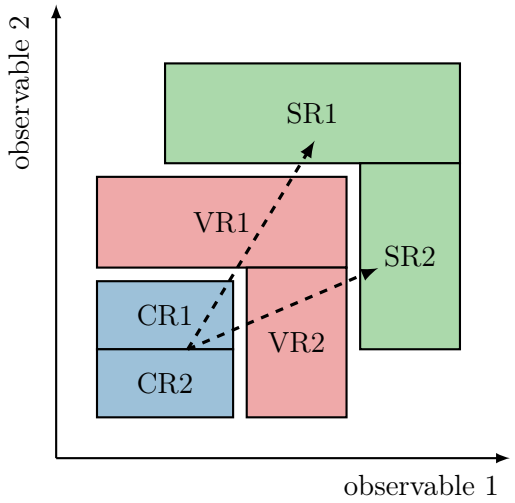


Figure 8.1: Schematic illustration of an analysis strategy with multiple control, validation and signal regions. The background extrapolation from the CRs to the SRs with its validation in the intermediate VRs is performed in the depicted two-dimensional phase space and is illustrated by the two dashed arrows.

8.1.3 Analysis Blinding and General Workflow

Once the initial background estimates in the CRs have been normalized via a maximum likelihood fit, as further explained in the next section, the fit results are extrapolated to the VRs in which the validity of the background estimation strategy is verified. If a satisfactory agreement between the predicted and observed numbers of events in the VRs is found, the backgrounds are further extrapolated to the SRs. Only at this stage of an analysis, when the full strategy is fixed with all regions defined and the backgrounds in the SRs are well understood and validated, the recorded data in the SRs is inspected and compared to the background predictions, a process usually referred to as *unblinding*. Keeping the SRs blinded until the full analysis strategy is fixed ensures objectivity by avoiding unintended biases by the analysis team and thus increases the confidence in the final analysis results. As explained in more detail in Section 8.2.3, the expected background yields, observed data events and predicted signal events in the SRs can then be compared and interpreted using statistical tests in order to draw conclusions about the validity of the hypothesized theory to be tested.
This kind of analysis workflow, as also implemented in the `HistFitter` [127] package to be utilized in the analysis presented in Chapter 9, is commonly used in particle physics data analyses and is schematically depicted in Fig. 8.2.

8.2 Statistics and Hypothesis Testing

This section introduces statistical methods and concepts commonly used in particle physics which allow for an interpretation of the observed data in the context of a theoretical model, and is largely based on Ref. [127] and Ref. [128]. Statistical tests can be used to quantify

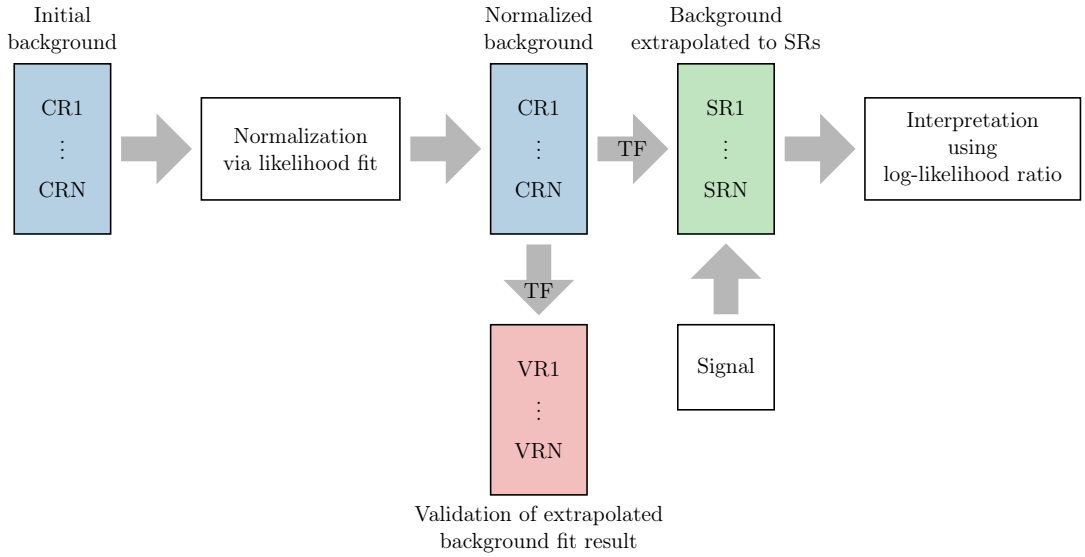


Figure 8.2: A typical analysis workflow in the search for new physics as implemented and explained in the `HistFitter` [127] package.

the agreement between the experimental observations and theoretical predictions and thus allow to make conclusive statements about the hypotheses to be tested.

8.2.1 The Likelihood Function

The statistical treatment begins with constructing a parametric model describing the observed data, modeled in terms of probability density functions (PDFs). In the simplest case of a pure counting experiment with a model, or *hypothesis*, predicting an average number of λ events, the corresponding PDF is given by the Poisson distribution

$$P(n|\lambda) = \frac{\lambda^n}{n!} e^{-\lambda}, \tag{8.3}$$

where n describes the number of actual observed events.
In particle physics analyses, the situation is typically more complex, as the numbers of predicted and observed events in all regions of interest, i.e. all SRs and CRs, are taken as input to construct a parametric model based on PDFs. The product of all Poisson distributions of event counts in the SRs and CRs is used to define the general *likelihood function* L of the analysis

$$L(\mu_{\text{sig}}, \vec{\mu}_{\text{bkg}}, \vec{\theta}) = \prod_{r \in \{\text{SRs,CRs}\}} P(N_r^{\text{obs}} | N_r^{\text{exp}}(\mu_{\text{sig}}, \vec{\mu}_{\text{bkg}}, \vec{\theta})) \times C_{\text{syst}}(\vec{\theta}), \tag{8.4}$$

a function of the parameters μ_{sig}, $\vec{\mu}_{\text{bkg}}$ and $\vec{\theta}$ only, treating the numbers of observed events N_r^{obs} as fixed at their observed values. The function parameters $(\mu_{\text{sig}}, \vec{\mu}_{\text{bkg}}, \vec{\theta})$ can be divided into two categories, the parameters of interest for the statistical interpretation μ_{sig} and $\vec{\mu}_{\text{bkg}}$, and the nuisance parameters (NP) $\vec{\theta}$, which are needed to define the statistical model but are not used in its final interpretation. Here, μ_{sig} denotes the so-called signal strength parameter and $\vec{\mu}_{\text{bkg}}$ the set of normalization factors μ_p for all relevant background

processes, as introduced in Eq. (8.1).
The set of nuisance parameters $\vec{\theta}$ is associated with the systematic uncertainties of the analysis: every single nuisance parameter θ_i contained in $\vec{\theta}$ describes a systematic uncertainty, interpolating between the nominal value with $\theta_i = 0$ and its variations with $\theta_i = \pm 1$, corresponding to the $\pm 1\sigma$ systematic uncertainty variations. The factor $C_{\text{syst}}(\vec{\theta})$ in Eq. (8.4) additionally constrains the systematic uncertainty estimate and is typically given by a product of standard normal distributions over all systematic uncertainties i:

$$C_{\text{syst}}(\vec{\theta}) = \prod_i \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\theta_i^2}{2}\right) \tag{8.5}$$

The expected number of events in region r, denoted by $N_r^{\text{exp}}(\mu_{\text{sig}}, \vec{\mu}_{\text{bkg}}, \vec{\theta})$ in Eq. (8.4), is given by

$$N_r^{\text{exp}}(\mu_{\text{sig}}, \vec{\mu}_{\text{bkg}}, \vec{\theta}) = \mu_{\text{sig}} \cdot N_{\text{r, sig}}^{\text{MC}}(\vec{\theta}) + \sum_p \mu_p \cdot N_{\text{r, p}}^{\text{MC}}(\vec{\theta}), \tag{8.6}$$

where $N_{\text{r, sig}}^{\text{MC}}(\vec{\theta})$ and $N_{\text{r, p}}^{\text{MC}}(\vec{\theta})$ describe the MC predicted signal yields and background yields for the SM process p in region r, respectively, while the sum runs over all background processes considered. Both of these quantities are dependent on the nuisance parameter set $\vec{\theta}$, as systematic variations can in general affect the acceptance and normalization of a process in a certain region of phase space, and thus influence the predicted rate of the given process in the region of interest. Furthermore, Eq. (8.6) demonstrates the role of the signal strength parameter μ_{sig} in the context of the statistical model: if $\mu_{\text{sig}} = 0$, the signal component is turned off and only background processes are considered in the contribution to the total expected event yield, while $\mu_{\text{sig}} = 1$ results in the nominally predicted signal expectations of the model under consideration [127].

8.2.2 Maximum Likelihood Estimation

Given an analysis' likelihood function $L(\vec{\xi})$, with the set of its model parameters abbreviated as $\vec{\xi} \equiv (\mu_{\text{sig}}, \vec{\mu}_{\text{bkg}}, \vec{\theta})$, a crucial step in an analysis is to find the values of the model parameters $\hat{\vec{\xi}}$ that maximize $L(\vec{\xi})$ for the given observations and thus lead to the best agreement between the model and observed data. The method employed to estimate these parameters is called the maximum likelihood estimation method and is, for computational reasons, usually based on maximizing the logarithm of the likelihood $\ln L(\vec{\xi})$ (or equivalently minimizing its negative logarithm) and proceeds with solving the system of equations

$$\frac{\partial \ln L}{\partial \xi_i} = 0 \tag{8.7}$$

for all model parameters i.
The maximum likelihood estimation of the best model parameters is most commonly referred to as the (maximum likelihood) *fit* of the parametric model to data.

8.2.3 Formalism of a Statistical Test

The statistical interpretation of the collected data, aiming to make conclusive statements about the validity of a theoretical model, is typically performed in terms of likelihood-based statistical tests. The following discusses important concepts and formalisms of statistical tests, expressed in the framework of frequentist statistics, and is largely based on Ref. [128].

Hypothesis Formulation

A statistical hypothesis test starts with the formulation of the hypothesis to be tested. In a search for new physics, the goal is typically to claim a discovery (or, if no excess in data can be found, an exclusion) of a hypothesized signal model, formulated in terms of two competing hypotheses, H_0 and H_1. In case of aiming to discover a new signal process, the so-called *null hypothesis* H_0 states that the observed data is purely described by already known SM processes and is thus also referred to as background-only hypothesis. H_0 is tested against the *alternative hypothesis* H_1, stating that the data is described by the sought-for signal in addition to the SM background processes. H_1 is therefore commonly referred to as the signal-plus-background hypothesis. The goal in discovering a new signal process is then to examine if the null hypothesis can be rejected by observed data in favor of the alternative hypothesis. If, on the other hand, in the absence of evidence of a new physics process exclusion limits need to be set, the signal-plus-background hypothesis is used as H_0 and tested against the background-only hypothesis H_1.
According to Eq. (8.6), a background-only hypothesis corresponds to a $\mu_{\text{sig}} = 0$ case, while the signal-plus-background hypothesis is given by setting the signal strength parameter to $\mu_{\text{sig}} = 1$. For convenience, the given hypotheses are therefore sometimes simplified to a more general notation and called the $\mu_{\text{sig}} = 0$ and $\mu_{\text{sig}} = 1$ hypotheses.

Test Statistics and Critical Region

In order to allow for a discrimination between the two competing hypotheses, a function of the measured data only, the so-called *test statistic* t, is defined. The test statistic can in general be understood as a metric of how much the observed data agrees or disagrees with the two stated hypotheses.
In order to test a hypothesized value of μ_{sig}, the *profile likelihood ratio*

$$\lambda(\mu_{\text{sig}}) = \frac{L(\mu_{\text{sig}}, \hat{\vec{\mu}}'_{\text{bkg}}, \hat{\vec{\theta}}')}{L(\hat{\mu}_{\text{sig}}, \hat{\vec{\mu}}_{\text{bkg}}, \hat{\vec{\theta}})} \tag{8.8}$$

is defined, where in the numerator $\hat{\vec{\mu}}'_{\text{bkg}}$ and $\hat{\vec{\theta}}'$ are the fitted maximum likelihood estimate parameter values for the specified signal strength μ_{sig}, and $\hat{\mu}_{\text{sig}}$, $\hat{\vec{\mu}}_{\text{bkg}}$ and $\hat{\vec{\theta}}$ in the denominator denote the full maximum likelihood estimate parameter set for the likelihood function L. The final test statistic t to be used for the statistical interpretation is now defined as

$$t = t_{\mu_{\text{sig}}} = -2 \ln \lambda(\mu_{\text{sig}}), \tag{8.9}$$

where high values of t correspond to greater incompatibility between data and the hypothesis μ_{sig} to be tested [128].
If the null hypothesis H_0 is accepted or rejected is now determined by where the observed value of t lies with respect to the so-called *critical region*, defined by a cut value t_{cut}, as exemplarily illustrated in Fig. 8.3(a). Here, $f(t|H_0)$ and $f(t|H_1)$ denote the probability density distributions of the test statistic t under the hypothesis of H_0 and H_1, respectively. The null hypothesis H_0 is rejected in favor of the alternative hypothesis H_1 if the observed value of t, t_{obs}, lies within the critical region with $t_{\text{obs}} > t_{\text{cut}}$.
The critical region also defines the so-called *significance level* α, the *confidence level* $1-\alpha$ as well as *power of the test* $1-\beta$. The quantity α, also known as the Type-I error rate, indicates the probability to reject H_0 while it is actually true and is illustrated by the blue area under the curve in Fig. 8.3(a). The Type-II error rate β is shown in red and states the probability to reject H_1 (and accept H_0 instead) while H_1 is true. In an ideal statistical test in the context of a particle physics discovery, both α and β are small, implying that the chance α of incorrectly claiming a new discovery (incorrectly accepting H_1 while H_0 is true) is small and the chance of missing a new discovery (i.e. incorrectly accepting H_0 while H_1 is true) is small as well.

The p-Value and CL$_s$ Construction

In order to quantify the level of agreement between a hypothesized value of μ_{sig} and observed data, a so-called p-value is computed. It is defined as

$$p_{\mu_{\text{sig}}} = \int_{t_{\mu_{\text{sig}},\text{obs}}}^{\infty} f(t_{\mu_{\text{sig}}}|\mu_{\text{sig}})dt_{\mu_{\text{sig}}}, \tag{8.10}$$

where $t_{\mu_{\text{sig}},\text{obs}}$ is the observed value of the test static and $f(t_{\mu_{\text{sig}}}|\mu_{\text{sig}})$ denotes the probability density function of $t_{\mu_{\text{sig}}}$ under the hypothesis of μ_{sig}. The value of $p_{\mu_{\text{sig}}}$, as illustrated in Fig. 8.3(b), can therefore be understood as the probability of finding data at least as extreme as the one observed, under the assumption that hypothesis μ_{sig} is true. In case of $\mu_{\text{sig}} = 0$, for example, the p_0-value is thus interpreted as the probability of finding data at least as extreme as the one observed, assuming that the background-only hypothesis is true. While the significance level of a test α is a pre-defined constant, a p-value is a function of the observed data and thus itself a random variable. If the observed p-value is smaller than the specified significance level, the incompatibility between data and the null hypothesis is significant enough to reject the null hypothesis and accept the alternative hypothesis instead.
In order to claim that a new physics process has been discovered, the background-only null hypothesis with $\mu_{\text{sig}} = 0$ needs to be rejected. By convention in the particle physics community, a value of $p_0 = 2.87 \times 10^{-7}$ or smaller is required to claim a discovery, while for the exclusion of a null hypothesis with signal strength μ_{sig} a less stringent threshold of $p_{\mu_{\text{sig}}} \leq \alpha = 0.05$ is chosen.
Often, p-values are expressed in terms of Gaussian significance Z, defined in such a way that the total upper tail probability of a Gaussian distributed variable located Z standard

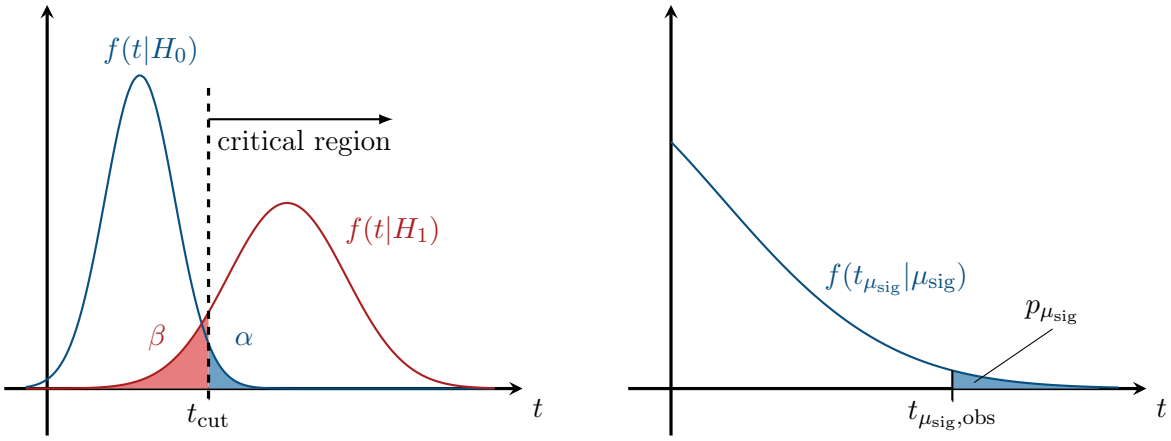


Figure 8.3: (a) Example probability density distributions f of a test static t under two different hypotheses H_0 and H_1 together with the threshold value t_{cut} defining the critical region. (b) Illustration showing the relation between an observed value of the test statistic $t_{\mu_{\text{sig}},\text{obs}}$ and its corresponding p-value $p_{\mu_{\text{sig}}}$ for an example probability density distribution $f(t_{\mu_{\text{sig}}}|\mu_{\text{sig}})$.

deviations above its mean value is equal to p:

$$Z = \Phi^{-1}(1-p) \tag{8.11}$$

Here, Φ^{-1} describes the inverse of the cumulative distribution of a standard Gaussian function. The above mentioned p-value conventions in the particle physics community for claiming discovery and exclusion thus translate to Gaussian significance value of $Z = 5$ and $Z = 1.64$, respectively [128].
When trying to exclude a certain hypothesis with signal strength μ_{sig} on the basis of the observed value of $p_{\mu_{\text{sig}}}$ as defined in Eq. (8.10), statistical downward fluctuations in data can easily lead to a premature, and thus unphysical, exclusion of the given hypothesis. The so-called CL_s metric [129], defined as

$$\text{CL}_s = \frac{p_{\mu_{\text{sig}}}}{1-p_0}, \tag{8.12}$$

is therefore often used instead of $p_{\mu_{\text{sig}}}$ in tests aiming to exclude certain signal models, as downward fluctuations in data will lead to larger p_0 values and the resulting larger CL_s values therefore avoid a premature exclusion of the model considered.

8.3 Common Fit Strategies

There are different fit strategies that are commonly employed in particle physics analyses. They differ in the combinations of SRs, CRs and VRs that are simultaneously used and by whether a signal model is considered or not. The three most commonly used fit strategies, the *background-only fit*, the *model-dependent signal fit* as well as the *model-independent signal fit*, will be discussed in the following, based on the explanations given in Ref. [127].

8.3.1 Background-only Fit

Aiming to estimate the total backgrounds in the VRs and SRs, this fit strategy considers only background samples without making any assumptions on a particular signal model. The fit is performed in the CRs only, which are assumed to be free of signal, normalizing the background processes to the observed CR event counts and thus allowing to extrapolate the background level predictions to the VRs and SRs. The background-only fit therefore allows for an unbiased comparison of the predicted event numbers with the ones actually observed in the SRs and VRs.

8.3.2 Model-dependent Signal Fit

The model-dependent signal fit is used when a specific signal model is studied. If the background-only fit does not show a significant excess in one or several SRs, this fit strategy allows to set exclusion limits on the studied signal model. If, on the other hand, an excess in the background-only fit was found, model properties such as the signal strength parameter μ_{sig} can be measured. This fit is simultaneously performed in the SRs and CRs while the signal samples are included in all regions in order to account for possible signal contamination in the CRs.

8.3.3 Model-independent Signal Fit

This fit strategy is independent of an underlying assumption of a signal model and is typically used to set upper limits on the number of events beyond the expected ones in the SRs. Similar to the model-dependent fit strategy, the fit is simultaneously performed in CRs and SRs, while no signal contamination in the CRs is considered. Also, the number of expected signal events in the SRs is used as a free parameter of the likelihood function in the fitting procedure. Since a specific shape of the signal is highly dependent on the chosen signal model, multiple bins in the SRs cannot be used and instead, the shape information is ignored during the fit by only using single-binned SRs.

Chapter 9

Search for Flavor-Violating Supersymmetry

9.1 Introduction

This chapter presents a search for stop quarks in a flavor-violating supersymmetric model, targeting a $\mathrm{tc} + E_{\mathrm{T}}^{\mathrm{miss}}$ final state using the full ATLAS dataset from the Run 2 data taking period with $139\,\mathrm{fb}^{-1}$ of pp collision data at $\sqrt{s} = 13\,\mathrm{TeV}$.
The goal of this analysis is to target specifically a largely unexplored area in stop-searches, in which the mixing between the second-generation and third-generation squarks results in a non-minimal flavor violation. Previous searches did not allow for any mixing between different squark generations and were therefore less sensitive in these regions of phase space, as discussed in Section 2.2.3.
After a short introduction to the target model and final state of interest in this section, the relevant Standard Model backgrounds will be discussed in Section 9.2. An overview of the data and Monte Carlo datasets used in this analysis will be given in Section 9.3, followed by the definition of the physics objects as well as main discriminating variables in Sections 9.4 and 9.5, respectively. The analysis strategy and applied preselections will be presented in Section 9.6, while the definition of SRs and the background estimation procedure will be discussed in Sections 9.7 and 9.8, respectively. Relevant systematic uncertainties will be covered in Section 9.9 and finally, the results and their interpretations will be presented in Section 9.10.

9.1.1 Target Model and Final State

The search presented in the following considers the simplified non-minimal flavor-violating SUSY model presented in Section 2.2.3 in a final state with a top quark, a charm quark and missing transverse momentum arising from the two undetected neutralinos.
The stop quarks are assumed to decay into a top and a neutralino, and into a charm and a neutralino with the same branching ratios $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) = \mathrm{BR}(\tilde{t}_1 \to \mathrm{c}\tilde{\chi}_1^0) = 50\%$. Only final states in which the mass difference between the stop and the neutralino is greater than the top mass, $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) > m_t$, and therefore only two-body decays of the stop, with a fully hadronic subsequent top decay are considered in the presented analysis. As the top

almost exclusively decays into a W boson and a bottom quark with BR(t $\rightarrow$ Wb) $\approx$ 100%, signal events are expected to contain a b-jet, a c-jet, $E_{\mathrm{T}}^{\mathrm{miss}}$ as well as additional jets arising from the hadronically decaying W, as illustrated in Fig. 9.1. Neither electrons nor muons (collectively referred to as leptons in the following) from a W decay are therefore expected in this 0-lepton (0L) final state.
The expected kinematics of the signal processes will largely depend on the two unknown parameters of the considered model, $m(\tilde{t}_1)$ and $m(\tilde{\chi}_1^0)$, and will have a great impact on the deployed analysis strategy discussed in Section 9.6.

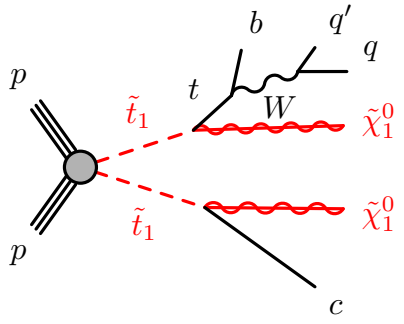


Figure 9.1: Pair production of stop quarks with subsequent two-body decays into a top quark, a charm quark and neutralinos. The full-hadronic decay channel of the top quark considered in this analysis is shown and the indication of antiparticles is implicit.

9.1.2 Contributions to the Analysis

The contributions made in the context of the given thesis to the analysis presented in the following include the definition and formulation of discriminating variables, as discussed in Section 9.5, which are used throughout the entire analysis in all SRs, CRs and VRs. Furthermore, significant contributions were made to the region definitions for the boosted top topologies (see A-type regions in the following), from which also the intermediate B-type regions were deduced. Additionally, contributions to the evaluation of systematic uncertainties as well as the final fit and their interpretations were made in the context of this thesis.

9.2 Relevant Standard Model Backgrounds Processes

Numerous Standard Model processes can result in final states with reconstructed b-jets, c-jets and missing transverse momentum and can therefore enter a signal-like phase space as background processes. The main relevant backgrounds for this analysis include Z+jets, W+jets, $\mathrm{t\bar{t}}$ as well as single-top processes and will be discussed in more detail in the following.
Other processes that can also potentially enter a signal-like phase space, such as diboson production, tWZ, tZ or $\mathrm{t\bar{t}}+\mathrm{X}$ (X = V, H) processes, have significantly lower cross sections and will be referred to as *Other* backgrounds in the following.

9.2.1 Z+Jets Backgrounds

One of the most dominant Standard Model background processes relevant for the presented search is the production of a Z boson in association with jets, referred to as Z+jets, as illustrated for some example production channels in Fig. 9.2.

Z bosons can decay into a pair of quarks, a pair of electrons, muons or taus as well as into a pair of neutrinos. The corresponding branching ratios for these decay modes are BR($Z \to q\bar{q}$) $\approx 69.9\%$, BR($Z \to e^+e^-, \mu^+\mu^-, \tau^+\tau^-$) $\approx 10.1\%$ and BR($Z \to \nu_e\bar{\nu}_e, \nu_\mu\bar{\nu}_\mu, \nu_\tau\bar{\nu}_\tau$) $\approx 20\%$ [130].

While the hadronic decays of the Z could results in b- or c-jets, the high missing transverse momentum requirement will not be reached as no invisible particles are present in this decay channel. Furthermore, Z decays into electrons and muons will most likely be rejected by the applied lepton veto in the 0L signal regions, while these events will be important for the definition of the 2L control region. The $Z \to \tau^+\tau^-$ channel with hadronically decays taus, however, will have a non-negligible mistag rate for tau jets (i.e. tau jets can be misidentified as b- or c-jets) and can simultaneously include invisible particles and therefore enter the signal region as a background process. Since these processes, however, only come with very small branching ratios, they only play a minor role in the total Z+jets background for this analysis.

The main Z+jets background arises from invisible Z decays into neutrinos, with additionally produced jets that can pass the b-tag and c-tag requirements. Also, even in the SRs requiring at least one top-tagged large-radius jet (see Section 9.7.2 and 9.7.3), Z+jets can be the dominant background due to the low rejection power and therefore significant mistag rate of the employed top-tagger (see Section 9.4.3).

9.2.2 W+Jets Backgrounds

While W bosons can be produced in association with jets in a very similar way as the Z+jets background processes, see again Fig. 9.2, they do not feature a fully invisible decay

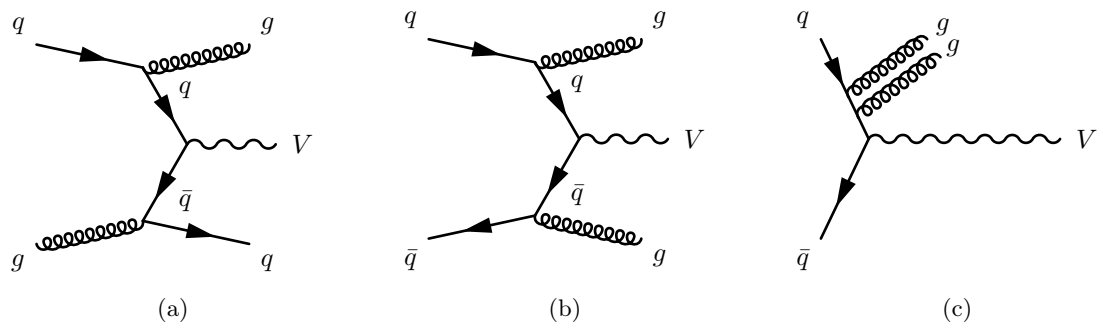


Figure 9.2: Examples of V = Z, W production processes in association with additional quark- and gluon initiated jets. Different quark flavors are not explicitly shown.

mode. Also, the hadronic decays into a pair of quarks $q\bar{q}'$ will again not reach the minimum required $E_{\mathrm{T}}^{\mathrm{miss}}$ threshold and the leptonic decays into $e\nu_e$ and $\mu\nu_\mu$, as well as into leptonically decaying taus, will most likely be rejected by the applied lepton veto in the SRs. The main contributions to the W+jets background therefore arises from $\mathrm{W} \to \tau\nu_\tau$ decays with hadronically decaying taus. These events can feature a high amount of $E_{\mathrm{T}}^{\mathrm{miss}}$ as well as c- and b-tagged jets, either stemming from a misidentified tau jet or ISR jets, and can therefore exhibit similar topologies as the sought-after signal processes.

9.2.3 $\mathrm{t\bar{t}}$ and Single-Top Backgrounds

Another important background for this analysis is the production of top quarks. These are either produced in pairs, as illustrated in Fig. 9.3 (a) and (b), or in so-called single-top events, as exemplarily shown in Fig. 9.3 (c) and (d).
As top quarks almost exclusively decay into a W boson and a bottom quark, the resulting final states of these events can again include hadronically decaying taus from the W decay, as explained above, and therefore involve invisible particles as well as mistagged tau jets or additional ISR jets, that allow these events to enter a SR-like phase space.

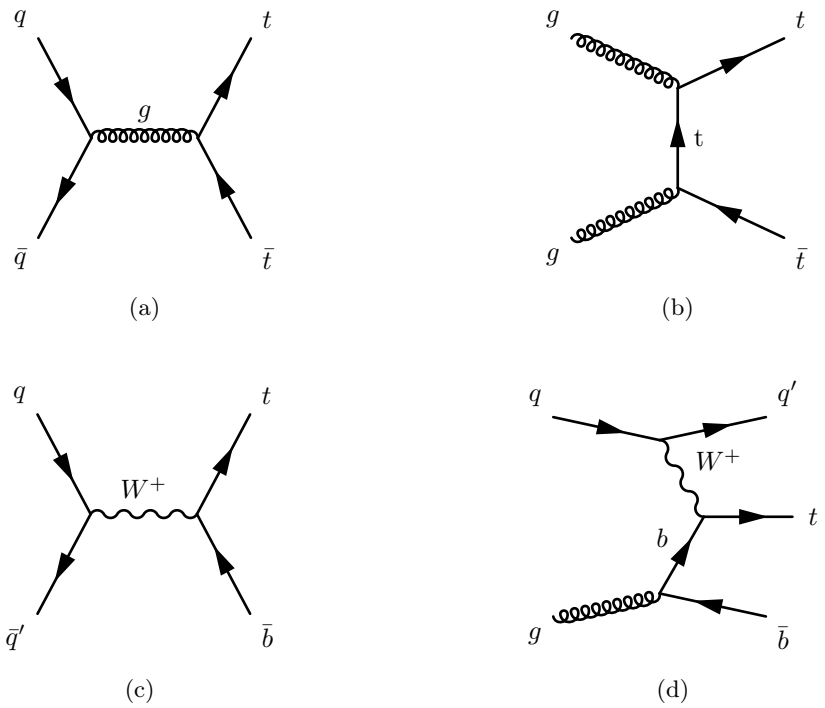


Figure 9.3: Examples of $\mathrm{t\bar{t}}$ production via s-channel (a) and t-channel (b) processes as well as of electroweak single-top production processes in (c) and (d).

9.3 Data and Monte Carlo Samples

9.3.1 ATLAS Run 2 Data

The datasets analyzed in the presented search were recorded by the ATLAS detector at the LHC in the Run 2 data taking period at a center-of-mass energy of $\sqrt{s} = 13\,\text{TeV}$. During this period, the LHC delivered a total integrated luminosity of $\mathcal{L}_{\text{int}} = 156\,\text{fb}^{-1}$, while the ATLAS recorded integrated luminosity amounts to $\mathcal{L}_{\text{int}} = 147\,\text{fb}^{-1}$, as was already shown in Fig. 3.2(b).

Only data recorded during stable LHC beam conditions and with all ATLAS subdetector systems fully operational are labeled *good for physics* and are used for physics analyses. This reduces the total integrated luminosity of the ATLAS Run 2 dataset used in this analysis to $\mathcal{L}_{\text{int}} = 139.0 \pm 2.4\,\text{fb}^{-1}$ [131].

Data recorded by ATLAS is split into different *runs*, each one typically containing data from a time period of several hours, which is further divided into so-called *luminosity blocks*, representing data collected over a time span of about one minute. All luminosity blocks that pass the good-for-physics data quality requirements are listed in so-called Good Run Lists (*GRLs)*. The GRLs used for the Run 2 datasets in this analysis are presented in Appendix B.1.

Events considered in this analysis are additionally required to have a reconstructed hard-scatter primary vertex with a least two associated tracks. The hard-scatter vertex is selected with the default method as the primary vertex with the largest $\sum p_{\text{T}}^2$, as explained in Chapter 5.

9.3.2 Signal Samples

The signal dataset for this analysis contains pair-produced stops, which subsequently decay with fixed and equal branching ratios of 50% to both $\tilde{t}_1 \to t\tilde{\chi}_1^0$ and $\tilde{t}_1 \to c\tilde{\chi}_1^0$. The resulting signal samples therefore consist of 25% pure $t\bar{t} + E_{\text{T}}^{\text{miss}}$ final states, 25% pure $c\bar{c} + E_{\text{T}}^{\text{miss}}$ final states as well as 50% mixed $tc + E_{\text{T}}^{\text{miss}}$ final states.

The samples are produced using the MadGraph5_aMC@NLOv2.8.1 generator [132], abbreviated with MadGraph 2.8.1 in the following, at next-to-leading order (NLO) with the NNPDF3.0NLO parton distribution function (PDF) set [133] for hard scatter matrix element (ME) calculations. MadGraph 2.8.1 is interfaced with Pythia 8.244 [83] with the Pythia A14 tune [134] to simulate stop decays, parton shower (PS), hadronization as well as the underlying event. The EvtGEN 1.7 generator [135] is used for modeling heavy-flavor decays at tree level, while an additional emission of up to two partons is allowed. The ME-PS matching is performed using the CKKW-L prescription [136] at a matching scale of $\frac{m(\tilde{t}_1)}{4}$. The cross sections of the signal processes are derived in next-to-next-to-leading-order (NNLO) in the strong coupling constant, while the resummation of soft gluon emissions is added in next-to-next-to-leading-logarithmic accuracy (NNLO+NNLL) [137–139]. The generators used for signal sample production are summarized in Table 9.1.

In order to ensure high signal statistics after applying analysis selections, a filter requirement at truth level of $E_{\text{T}}^{\text{miss}} > 100\,\text{GeV}$ is applied during signal sample generation, while

the subsequent full ATLAS detector simulation is performed using GEANT4 [88]. The signal samples are produced in the MC16a, MC16d and MC16e campaigns, dedicated to the data taking periods in the years 2015+2016, 2017 and 2018, respectively. In order to match the MC pile-up distributions to the ones found in data, each campaign is scaled with a dedicated pile-up profile and re-weighted to account for the different integrated luminosities in the respective years.

As both $m(\tilde{t}_1)$ and $m(\tilde{\chi}_1^0)$ are free parameters of the signal model, in principle all possible combinations have to be tested. Since this is clearly not feasible, in practice these two parameters are systematically varied to create a grid of 94 different signal model scenarios, covering a wide range in the experimentally accessible parameter space. Signal scenarios with stop masses of $400\,\text{GeV} \leq m(\tilde{t}_1) \leq 1300\,\text{GeV}$ and neutralino masses of $m(\tilde{\chi}_1^0) \leq 700\,\text{GeV}$ are considered and result in the grid of signal points that is shown in Fig. 9.4, together with the corresponding signal production cross sections.

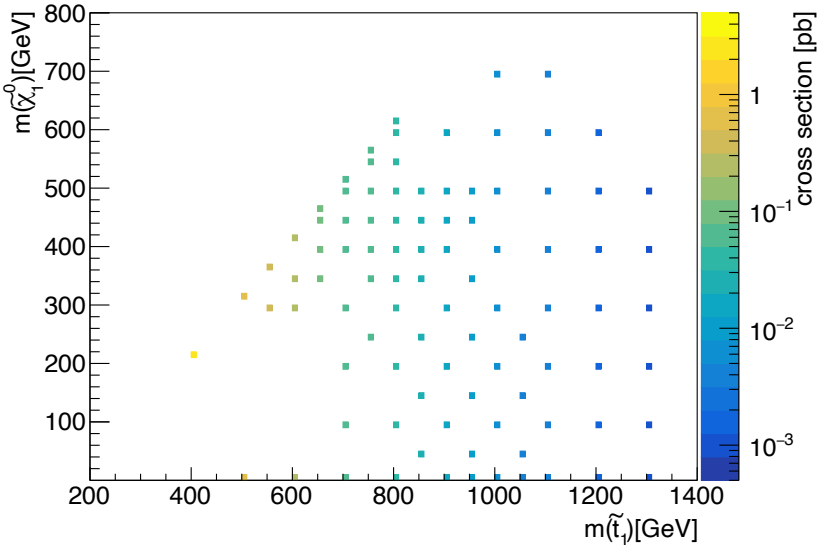


Figure 9.4: Overview of the signal grid used in this analysis and the corresponding signal production cross sections.

9.3.3 Background Samples

Similar to the signal model production described above, dedicated Monte Carlo samples are simulated in order to model the relevant SM backgrounds affecting the presented analysis.

While the nominal $t\bar{t}$ and single-top processes are generated by the POWHEG-BOX [140] event generator with the NNPDF2.3LO PDF set, interfaced with PYTHIA 8.244 for parton showering, the remaining top processes are simulated by MADGRAPH 2.3.3 together with PYTHIA 8.244. V+jets and diboson events are generated using SHERPA 2.2.1 and SHERPA 2.2.1/2.2.2, respectively, together with the NNPDF3.0NNLO PDF set. SHERPA uses a five-flavor-number scheme (5FNS) with massless charm and bottom quarks and models V production processes in association with up to two additional jets in NLO ME

calculations, processes with three and four jets at LO ME calculations, while even higher jet multiplicities are covered by the parton showering.
A detailed summary of the deployed generators together with the PDFs and used parameter sets for the nominal background samples is listed in Table 9.1. All nominal background samples are produced using the full `Geant4` ATLAS detector simulation, while additional samples used for the evaluation of theoretical uncertainties of the $t\bar{t}$ and single-top backgrounds are produced with the `Atlfast-II` [141] fast simulation package.

Process	Matrix element	Parton shower	PDF set	Tune	Cross section
$tc + E_T^{miss}$	MadGraph 2.8.1	Pythia 8.244	NNPDF3.0NLO	A14	NNLO+NNLL
$t\bar{t}$	Powheg-Box [140]	Pythia 8.244	NNPDF2.3LO	A14	NNLO+NNLL [142, 143]
Single-top	Powheg-Box	Pythia 8.244	NNPDF2.3LO	A14	NNLO+NNLL [144–146]
$t\bar{t} + X$	MadGraph 2.3.3	Pythia 8.244	NNPDF3.0NLO	A14	NLO [132]
tZ	MadGraph 2.3.3	Pythia 8.244	NNPDF2.3LO	A14	LO
tWZ	MadGraph 2.3.3	Pythia 8.244	NNPDF2.3LO	A14	NLO
V+jets	Sherpa 2.2.1 [84]		NNPDF3.0NNLO	Default	NNLO [147]
VV	Sherpa 2.2.1/2.2.2		NNPDF3.0NNLO	Default	NLO

Table 9.1: Overview of the generator configurations used for simulating the $tc + E_T^{miss}$ signal events as well as the nominal background samples with $V = Z, W$ and $X = V, H$. For conciseness, the generator MadGraph5_aMC@NLO is abbreviated with MadGraph.

9.4 Object Definitions and Selections

The reconstruction, identification and calibration of various different physics objects used in this analysis, including electrons, muons, (heavy-flavor) jets and E_T^{miss}, will be discussed in the following.

9.4.1 Electrons

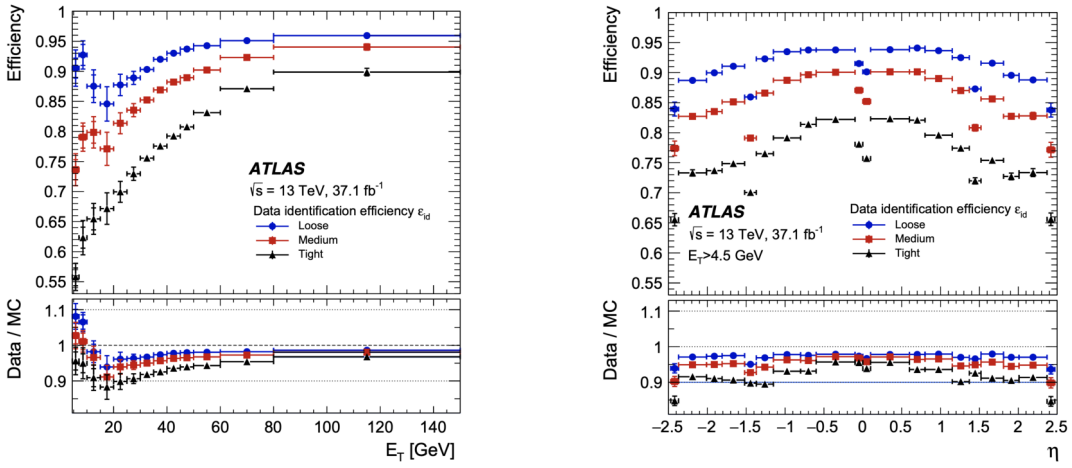
As electrons and positrons, in the following collectively referred to as electrons, often radiate bremsstrahlung photons when traversing detector material, which may further convert into electron-positron pairs that in turn can again interact with detector material, one primary electron often results in a collimated set of multiple charged particle tracks that can be reconstructed and matched to a single electromagnetic calorimeter energy cluster. The reconstruction and identification of electrons is thus based on reconstructed energy clusters in the electromagnetic calorimeter that can be associated to charged particle tracks in the Inner Detector, as explained in more detail in the following [148].
In a first step, a sliding-window approach with an $(\eta \times \phi)$ size of (3×5) calorimeter towers is used to find localized energy clusters in the ECAL with total energy deposits greater than 2.5 GeV, which subsequently serve as cluster seeds to match reconstructed ID tracks. ID tracks within the considered $\eta \times \phi$ seed cluster region with at least four silicon hits undergo

a refit procedure using the Gaussian Sum Filter technique [149], a generalization of the Kalman filter method explained in Chapter 4 that more efficiently accounts for material-induced energy losses by considering the non-linear effects related to bremsstrahlung.
Refitted charged particle tracks that match an ECAL seed cluster are considered an electron candidate, whereas seed clusters for which no compatible refitted ID track exist are labeled as unconverted photon. Furthermore, energies of reconstructed electron candidates are calibrated to match the original electron energies using multivariate methods described in Ref. [150].
In a last step, electron candidates with $|\eta| < 2.47$ are evaluated using a likelihood-based identification method in order to efficiently identify and select real electrons originating from the hard-scatter vertex [148].
Three likelihood-based electron identification working points, called `Loose`, `Medium` and `Tight`, are defined, differing in their signal identification efficiency and background rejection power. The electron identification efficiency in $Z \to ee$ events is shown in Fig. 9.5 for these three different working points as a function of E_T and η.
In the presented analysis, *baseline electrons* are required to satisfy $p_T > 4.5\,\text{GeV}$, $|\eta| < 2.47$ and $|z_0 \sin\theta| < 0.5\,\text{mm}$, as well as to pass the `LooseAndBLayerLLH` identification, a slight variation of the above mentioned `Loose` working point, as explained in more detail in Ref. [148]. *Signal electrons* are additionally required to pass the `TightLLH` identification (see again Ref. [148]) with $p_T > 10\,\text{GeV}$. Furthermore, if $p_T < 200\,\text{GeV}$, electrons are also required to fulfill the `FCLoose` [148] *isolation* requirement, or the `FCHighPtCaloOnly` [148] requirement otherwise. The use of isolation variables generally helps to efficiently differentiate signal objects originating from the hard-scatter vertex or the decay of heavy resonances from those objects originating from background processes such as conversions in the detector material. While signal objects are typically more isolated, featuring only very little activity surrounding the object in the ID and calorimeter system, objects originating from background processes often show much higher activities in their close vicinity.
Lastly, a transverse impact parameter significance below 5 is required for the electron-associated reconstructed track to ensure a high compatibility with the hard-scatter vertex.

9.4.2 Muons

The reconstruction and identification process of muons begins with separately reconstructing tracks in the ID and MS, which can subsequently be combined to form the muon objects used in analyses. While the ID track reconstruction is performed as described in Chapter 4, the muon reconstruction in the MS will be briefly discussed in the following [151].
In a first step of the muon reconstruction in the MS, a hit pattern finding approach is employed in each of the muon chambers to form individual track segments. Compatible track segments from different MS layers are subsequently fit together to form muon candidates, while at least two matching segments are required in this process[1]. All hits associated to a muon track candidate are used to perform a global χ^2 fit, while outlier hits with too

[1] In the transition region between the barrel and end-cap systems a single track segment is enough to form a muon candidate.



(a) (b)

Figure 9.5: Electron data identification efficiency in Z → ee events for `Loose`, `Medium`, and `Tight` working points as a function of (a) E_{T} and (b) η [148].

large individual contributions to the total χ^2 are removed and the fit is repeated. Also, the trajectory is refit if additional compatible hits are found and added to the muon track candidate.

Dedicated algorithms are employed to perform the combined ID-MS muon reconstruction, based on information provided by the ID, MS as well as the calorimeters. Depending on the sub-detector systems used for the muon reconstruction, four different muon types are defined [151]:

- So-called *combined* (CB) muons are defined by independently performing track reconstruction in the ID and MS, while both inside-out as well as outside-in pattern recognition approaches, in which the reconstructed ID track is extrapolated outward and matched to the MS track and vice versa, are used to build the full muon tracks.
- Muons are classified as *segment-tagged* (ST) if an extrapolated ID track is associated with at least one local track segment in the muon chamber, which can occur if the muon, e.g. due to low transverse momentum, only crossed one single MS chamber layer.
- Especially used in regions where the MS is only partially instrumented, a *calorimeter-tagged* (CT) muon is defined if an ID track can be associated to calorimeter energy deposits that are compatible with a minimum-ionizing particle.
- *Extrapolated* (ME) muons are mainly used in the region not covered by the ID with $2.5 < |\eta| < 2.7$ and are exclusively based on the reconstructed trajectory in MS without any ID information.

Possibly occurring ambiguities, in which the same ID track is simultaneously used by different muons types, are resolved by given preference to muons types in the following order: CB, ST, CT.

Similar to the above described identification of electrons, different muon identification working points are defined based on the application of dedicated quality requirements (see Ref. [151] for more details). Four muon identification working points, called `Loose`, `Medium`, `Tight` and `High-`p_{T}, are defined, differing in their identification efficiency and rejection power against backgrounds that mimic muon signatures. The efficiency of the muon reconstruction as a function of the transverse momentum is shown in Fig. 9.6 for the `Medium` muon selection.

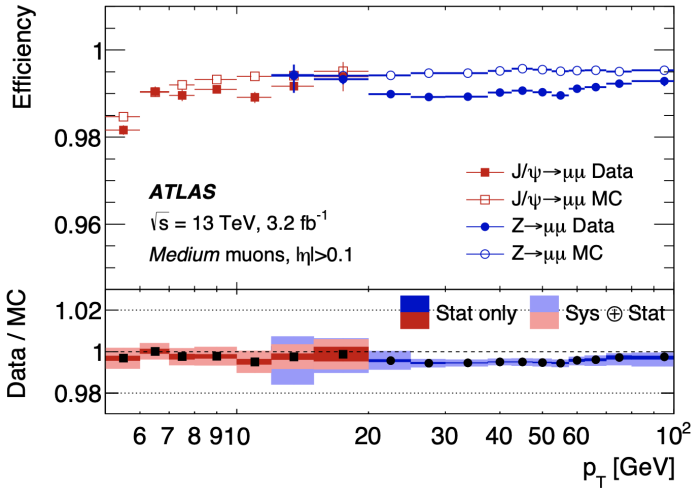


Figure 9.6: Reconstruction efficiency for `Medium` muons as a function of the muon p_{T} for MC and data collected in the year 2015 [151].

Baseline muons used in the presented analysis are required to satisfy $p_{\mathrm{T}} > 4\,\mathrm{GeV}$, $|\eta| < 2.7$ and $|z_0 \sin\theta| < 0.5\,\mathrm{mm}$ as well as to pass the `Medium` muon selection. Similar to signal electrons, *signal muons* are additionally required to pass the `FCLoose` isolation requirement as well as a $p_{\mathrm{T}} > 10\,\mathrm{GeV}$ threshold, while the muon-associated reconstructed track needs to show a high compatibility with the hard-scatter vertex with a transverse impact parameter significance below 3.

9.4.3 Jets

The reconstruction of particle jets in ATLAS can be performed using a variety of different input objects, such as ID tracks, calorimeter energy deposits or a combination of both. As the presented analysis makes use of the two latter approaches, only these will be covered in more detail in the following. The jet finding process starts with the formation of topological clusters of calorimeter cells, which can directly be used as input for jet reconstruction algorithm. Alternatively, so-called particle flow objects can be built based on these topological calorimeter clusters and additional ID track information, which are in turn used as input for a dedicated jet finding algorithm to reconstruct the jet objects of interest, as explained in the following [152].

Topological Calorimeter Cell Clustering

A three-dimensional topological clustering algorithm [153, 154] is used to group neighboring calorimeter cells with significantly high energy depositions into clusters, attempting to extract signal measurements while reducing the contamination from background processes, such as electronic detector noise or fluctuations due to pile-up effects. In a first step of the topological-cluster (*topo-cluster*) formation, so-called *seed cells* are identified as cells with a very significant signal-to-noise ratio t above a high threshold $t_{\text{seed}} = 4$, where the signal is given by the absolute value of the measured energy deposition in the respective calorimeter cell and noise includes electronics and expected pileup contributions. All neighboring cells, defined as the eight calorimeter cells that are directly adjacent to the seed cell within the same detector layer or optionally also including cells from neighboring detector layers with at least a partial overlap in the (η, ϕ) plane, are now considered for the clustering process. Neighboring cells are included into the cluster if their signal-to-noise ratio is above the threshold $t_{\text{neighbor}} = 2$ and the process continues iteratively until all topo-clusters are formed.

Particle Flow Algorithm

With the knowledge of the reconstructed topo-clusters, a jet finding algorithm, purely based on calorimeter information, can already be employed at this stage. However, combining the knowledge of calorimeter energy depositions with tracking information provided by the ID in a so-called Particle Flow algorithm [155] can enhance the subsequent jet finding performance, as two complementary systems with different strengths in different regimes are utilized: while the tracking system provides high momentum and angular resolution, specifically for low-p_{T} particles, along with great pile-up suppression capabilities, the calorimeter's advantage is the reconstruction of neutral and high-p_{T} particles as well as its greater acceptance in the forward region.
Particle Flow Objects (PFOs) are created by matching well-measured tracks to one or multiple topo-clusters. For each single track, the expected deposited energy in the calorimeter is computed based on the track's momentum, and the probability of the track creating multiple topo-clusters in the calorimeter is evaluated. The expected track's energy depositions are removed from the matched topo-clusters and if the remaining energy values are consistent with expected shower fluctuations, the created track-topo-cluster system can be considered a PFO, which is used as input for the following jet finding procedure.

Anti-k_t Jet Finding

Jet finding in ATLAS is performed using the Anti-k_t jet clustering algorithm [156], which is agnostic to the exact input type and can therefore be applied on various different objects, such as particle tracks, topo-clusters or, as done for the reconstruction of small-radius jets used in this analysis, on PFOs.
The Anti-k_t algorithm is a sequential recombination algorithm that iteratively clusters objects which are close to one another according to a well-defined distance measure. The

algorithm makes use of the following distance definitions

$$d_{ij} = \min\left(p_{T,i}^{2p}, p_{T,}^{2p} j\right) \frac{\Delta_{ij}^2}{R^2},$$
$$d_{iB} = p_{T,i}^{2p}$$

between objects i and j, where $p_{T,i}$ and $p_{T,j}$ denote the transverse momenta of objects i and j, respectively, and $\Delta_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$ with the rapidity y_i and azimuth ϕ_i of object i. R is the radius parameter of the algorithm and p governs the relative power between the energy and distance scale Δ_{ij}, chosen to be $p = -1$ in the Anti-k_t algorithm. The algorithm iteratively proceeds as follows, until no objects are left for further clustering:

1. Determine distances d_{ij} and d_{iB} for all i, j.
2. If a d_{ij} distance is the smallest, combine corresponding objects i and j into a new, clustered object.
3. If for any i the distance d_{iB} is the smallest, define object i as a jet and remove all associated objects.

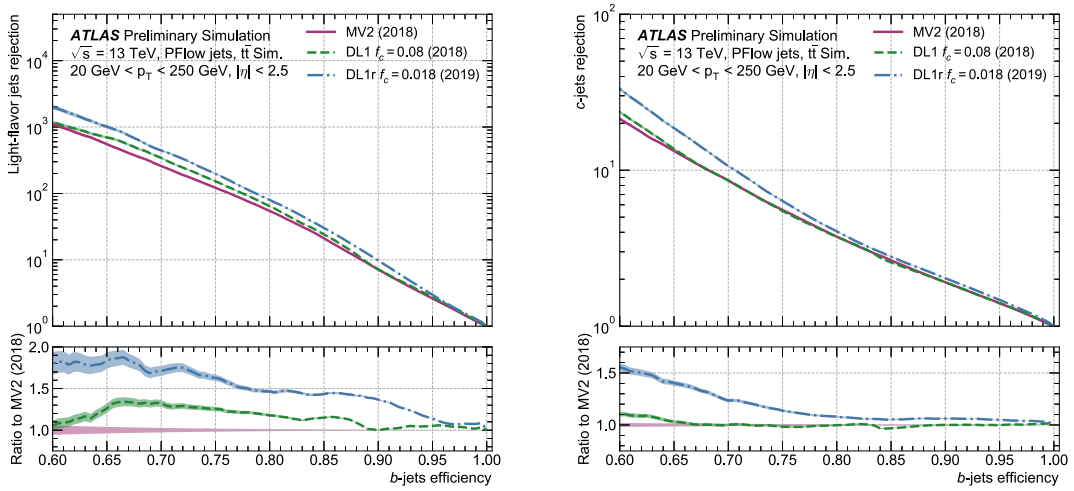
The presented analysis makes use of small-radius jets with $R = 0.4$ as well as large-radius jets, especially targeting boosted top topologies, with $R = 1.0$. While the small-radius jets are reconstructed on the basis of PFOs, the reconstructed large-radius jets use pure calorimeter topo-cluster information as input.
Since the energy depositions in the calorimeter cells are measured at an electromagnetic scale, a local cell weighting (LCW) calibration can be performed in order to account for differences in the detector response between electromagnetically and hadronically interacting particles [152]. A jet energy scale (JES) calibration is used to match the energy scale of reconstructed jets to the one of the originally simulated truth jets and includes for instance origin corrections, changing the jet direction to point to the reconstructed primary hard-scatter vertex, pile-up corrections and absolute detector response corrections [157]. Moreover, undesired pile-up jets are suppressed using multivariate methods, such as the Jet Vertex Tagger (JVT) technique [158, 159].
The presented analysis defines small-radius *baseline jets* as PFO-based $R = 0.4$ Anti-k_t jets with $p_T > 20\,\text{GeV}$ and $|\eta| < 2.8$. *Signal jets* are additionally required to pass a `Tight` JVT-based pile-up rejection score of $s_{JVT} > 0.5$. Large-radius jets are defined as topo-cluster-based $R = 1.0$ Anti-k_t jets, calibrated using an LCW calibration scheme. The resulting large-radius jet collection is further trimmed to reduce effects like pileup contamination by first reclustering all jet constituents into $R = 0.2$ sub-jets and removing all underlying objects with p_T lower than 5% of the total jet transverse momentum. Dedicated calibration methods similar to the ones applied for small-radius jets are subsequently used to correct various jet quantities such as its momentum and mass (see Ref. [160]).
The uncertainties related to the applied calibration techniques, such as the JES calibration or also the calibration of the jet energy resolution (JER), are some of the important systematic experimental uncertainties further discussed in Section 9.9.

B-tagged Jets

Small-radius signal jets are further evaluated to determine the flavor of the quark it originated from. As already discussed in Section 5.5 in the context of secondary vertex reconstruction and heavy-flavor tagging algorithms, dedicated b-tagging methods are employed in ATLAS, including the `MV2` [116], the `DL1` [116] as well as the `DL1r` [116, 117] algorithm, aiming to identify b-initiated jets while trying to keep the contamination from c-initiated or light-flavor jets as low as possible.

The probability of identifying a truth b-initiated jet as a b-jet is given by the b-tagging efficiency ϵ_b, while the probabilities of identifying a truth c-jet or a light-jet as a b-jet are given by ϵ_c and ϵ_u, respectively. Instead of quoting an efficiency for misidentifying a c- or light-jet as a b-jet, the c-jets rejection $1/\epsilon_c$ and light-jets rejection $1/\epsilon_u$ factors are usually used. If, for example, the probability to identify a truth c-jet as a b-jet is given by $\epsilon_c = 20\%$, the resulting c-jets rejection corresponds to $1/\epsilon_c = 5$, meaning that one out of five truth c-jets will be misidentified as a b-jet.

The light-flavor jets rejection as well as the c-jets rejection as a function of the b-jets efficiency are shown in Fig. 9.7 for the `MV2`, `DL1` and `DL1r` b-tagging algorithms. This analysis makes use of the DL1r algorithm at a recommended $\epsilon_b = 77\%$ b-tagging working point, corresponding to a light-jet rejection of $1/\epsilon_u = 170$ as well as a c-jet rejection of $1/\epsilon_c = 5$.



(a) (b)

Figure 9.7: Comparison of (a) the light-flavor jets rejection and (b) the c-jets rejection as a function of the b-jets efficiency for the `MV2`, `DL1` and `DL1r` flavor-tagging algorithms, evaluated on PFO-based $R = 0.4$ Anti-k_t jets with $20\,\text{GeV} < p_\text{T} < 250\,\text{GeV}$ and $|\eta| < 2.5$ [161].

C-tagged Jets

In order to identify c-initiated jets, a c-tagging dedicated adaptation of the `DL1r` algorithm was developed in the context of this analysis and is based on interchanging the b-jet and c-jet probabilities p_b and p_c in the original definition of the `DL1r` score, as already presented

in Eq. (5.58), defining the $\texttt{DL1r}_c$ score as

$$\texttt{DL1r}_c = \ln\left(\frac{p_c}{f_b \cdot p_b + (1 - f_b) \cdot p_u}\right), \tag{9.1}$$

where again both the cut on $\texttt{DL1r}_c$ as well as f_b are adjustable parameters of the algorithm that can be used to tune the desired c-tagging efficiency, and p_b, p_c and p_u denote the neural network-predicted probabilities of a jet to be a b-jet, c-jet or light-jet, respectively. Since a jet could in principle pass both the DL1r and $\texttt{DL1r}_c$ requirements, and would therefore be tagged as both a b-jet and a c-jet, it is important to define the order of applied tagging operations. As the b-tagging algorithm is generally more efficient than the c-tagging, it was decided to first employ the b-tagging algorithm and subsequently only apply the c-tagger to the jets that failed the b-tagging requirements. This *c-tagging with b-veto* approach is schematically illustrated in Fig. 9.8.

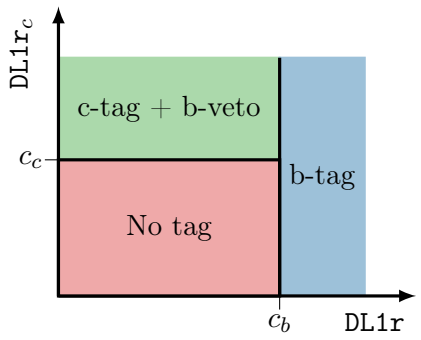


Figure 9.8: Schematic illustration of how the DL1r and $\texttt{DL1r}_c$ output scores and their respective cut values c_b and c_c are used to define b-tagged jets, c-tagged with b-veto jets as well as non-tagged jets.

Since the DL1r neural network is specifically optimized for the classification of b-jets, using the modified $\texttt{DL1r}_c$ score can result in high light-jet mistag rates if high c-jet efficiencies are desired. Thus, in order to keep the mistag rate as low as possible, the tagger is optimized for a target efficiency of $\epsilon_c = 20\%$, which is achieved by choosing the two configurable parameters of the $\texttt{DL1r}_c$ algorithm as $f_b = 0.28$ and $\texttt{DL1r}_c > 1.35 = c_c$. Fig. 9.9 shows the algorithm's light-jet rejection as a function of the b-jet rejection for three different c-tagging efficiencies. The working point optimized for this analysis corresponds to the point on the $\epsilon_c = 20\%$ efficiency curve with b-jet rejection of $1/\epsilon_b = 30$ and light-jet rejection of $1/\epsilon_u = 60$.

Top-tagged Jets

Certain decay topologies with very collimated decay products along the direction of the decaying mother particle might be reconstructed as large-radius jets, which can be used to identify the nature of the originally decaying mother particle. This analysis employs a neural network-based top-tagging algorithm [162, 163], which is trained to identify top-initiated $R = 1.0$ jets based on various different inputs such as the jet mass and jet substructure variables. Only large-radius jets with transverse momenta $350\,\text{GeV} < p_\text{T} <$

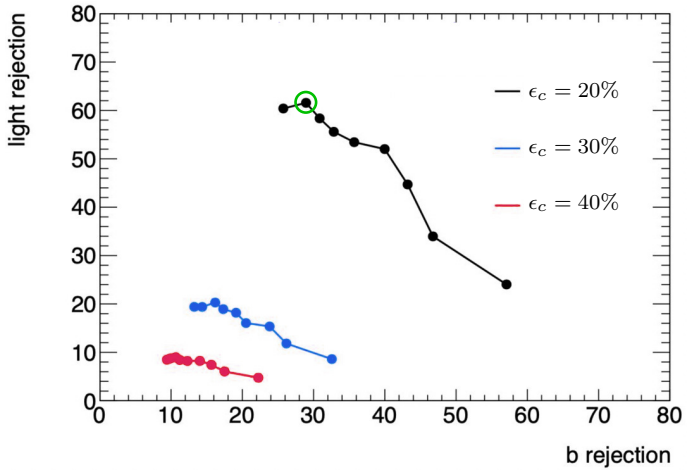


Figure 9.9: Light-jet rejection as a function of b-jet rejection for different c-tagging efficiencies. The c-tagging algorithm used in the presented analysis is optimized for the working point with b-jet rejection $1/\epsilon_b = 30$ and light-jet rejection $1/\epsilon_u = 60$ on the black $\epsilon_c = 20\%$ efficiency curve as indicated by the green circle.

2500 GeV and jet masses 40 GeV $< m_j <$ 600 GeV are considered for top-tagging, as recommended for the employed algorithms. Fig. 9.10 shows the background rejection $1/\epsilon_{\text{bkg}}$ as a function of the jet p_{T} for different top-tagger configurations. In order to retain enough statistics when defining signal and control regions later on, and given the already low c-tagging efficiency of $\epsilon_c = 20\%$, this analysis uses the inclusive $\epsilon_{\text{sig}} = 80\%$ top-tagger configuration, illustrated with a dashed blue line. The high top-tagging efficiency entails a very low background rejection power below $1/\epsilon_{\text{bkg}} = 10$ across a wide jet p_{T} range, leading to a large proportion of jets that are not initiated by top quarks being misidentified as top-jets.

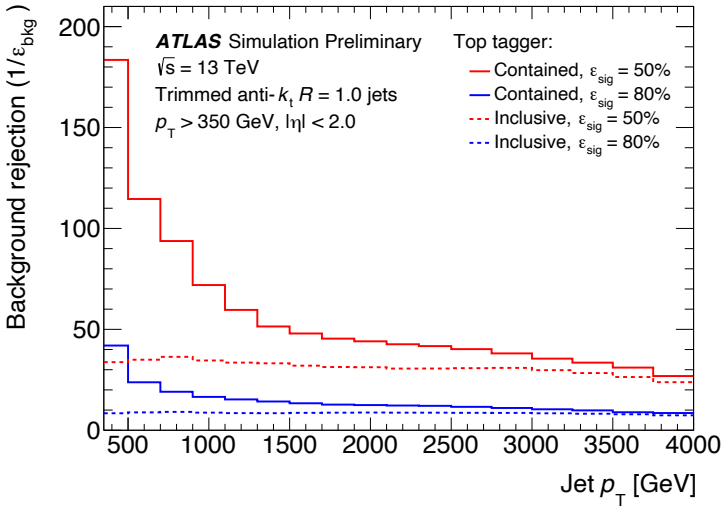


Figure 9.10: Background rejection in multi-jet events for different top-tagger configurations as a function of the jet p_{T}. The presented analysis makes use of the inclusive $\epsilon_{\text{sig}} = 80\%$ version (dashed blue line) [163].

9.4.4 $E_{\mathrm{T}}^{\mathrm{miss}}$

The missing transverse momentum reconstruction is based on measuring individual $E_{x,y}^{\mathrm{miss}}$ components, which can subsequently be used to define the total $E_{\mathrm{T}}^{\mathrm{miss}}$ magnitude and its azimuthal angle ϕ^{miss}.
The $E_{x,y}^{\mathrm{miss}}$ components are given by

$$E_{x,y}^{\mathrm{miss}} = E_{x,y}^{\mathrm{miss},e} + E_{x,y}^{\mathrm{miss},\gamma} + E_{x,y}^{\mathrm{miss},\tau} + E_{x,y}^{\mathrm{miss,\,jets}} + E_{x,y}^{\mathrm{miss},\mu} + E_{x,y}^{\mathrm{miss,\,soft}}, \tag{9.2}$$

where each individual term is defined as the negative vectorial sum of the respective object momenta. Selected and calibrated electrons (e), photons (γ), hadronically decaying taus (τ), jets as well as muons (μ) are considered, while the soft terms refers to all other reconstructed objects that need to be included, such as single ID tracks or individual calorimeter signals [164].
The total $E_{\mathrm{T}}^{\mathrm{miss}}$ value as well as its azimuthal direction are now given by

$$E_{\mathrm{T}}^{\mathrm{miss}} = \sqrt{(E_x^{\mathrm{miss}})^2 + (E_y^{\mathrm{miss}})^2}, \tag{9.3}$$

$$\phi^{\mathrm{miss}} = \arctan\left(E_y^{\mathrm{miss}}/E_x^{\mathrm{miss}}\right). \tag{9.4}$$

The default `Tight` [164] $E_{\mathrm{T}}^{\mathrm{miss}}$ working point is used in this analysis.
Another $E_{\mathrm{T}}^{\mathrm{miss}}$-based quantity used in the presented analysis is the object-based $E_{\mathrm{T}}^{\mathrm{miss}}$ significance [165], defined in a reference frame in which the x-y coordinate system is rotated parallel and perpendicular to the total $E_{x,y}^{\mathrm{miss}}$ direction as

$$\mathcal{S} = \frac{E_{\mathrm{T}}^{\mathrm{miss}}}{\sqrt{\sigma_{\parallel}^2(1-\rho^2)}}, \tag{9.5}$$

where $\sigma_{\parallel}^2$ denotes the total variance in parallel direction to $E_{x,y}^{\mathrm{miss}}$ and ρ^2 is a correlation factor between the parallel and perpendicular measurement components.
The object-based $E_{\mathrm{T}}^{\mathrm{miss}}$ significance can be advantageous for better discriminating between background events, where $E_{\mathrm{T}}^{\mathrm{miss}}$ arises from poorly measured objects, and signal events with truly invisible particles in the final state. High values of $\mathcal{S}$ suggest that the observed $E_{\mathrm{T}}^{\mathrm{miss}}$ cannot be explained by mismeasured objects or poor momentum resolution, and instead indicate that the event most likely contains undetected objects.

9.4.5 Overlap Removal

The above discussed physics objects definitions are not exclusive and can therefore lead to overlaps, in which a single true physics object is identified as two different reconstructed objects. In order to resolve these ambiguities, an *overlap removal* technique on baseline objects is applied in this analysis:

- **Muon-electron overlaps:** If a calorimeter-tagged muon and an electron share the same ID track, the muon is rejected in favor for the electron. For all other muon

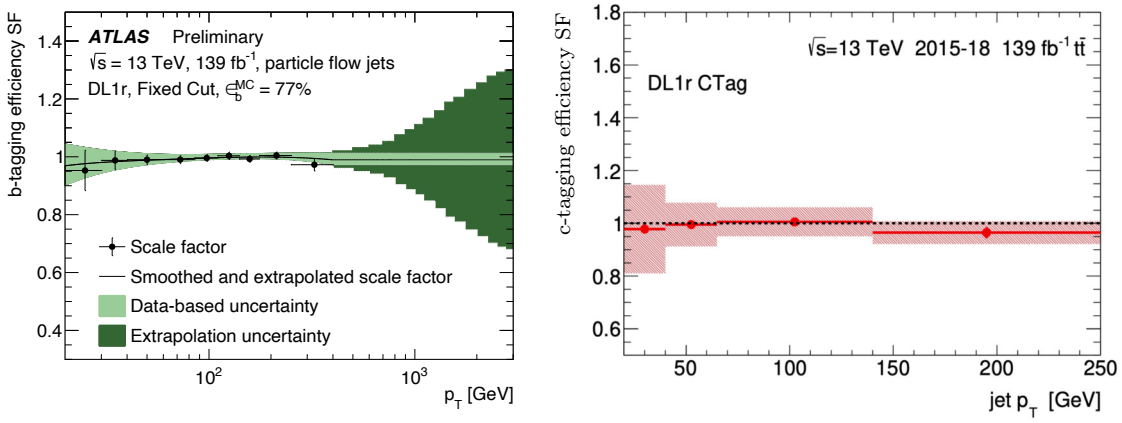
object types that share an ID track with an electron, the muon is kept while the electron is rejected.

- **Jet-electron overlaps:** If a jet and an electron overlap with $\Delta R < 0.2$ and the jet is not b-tagged and fulfills $p_{\mathrm{T}} < 100\,\mathrm{GeV}$, the jet is rejected in favor of the electron. The jet will be kept in all other cases. For all other jets and electrons with $\Delta R < 0.4$ the electron is rejected while the jet is kept.
- **Jet-muon overlaps:** If a jet with less than three tracks overlaps with a muon with $\Delta R < 0.2$, the jet is removed and the muon is kept. For all other jets and muons with $\Delta R < 0.4$ the muon is removed while the jet is kept.

9.4.6 Scale Factors

In order to calibrate MC to match data, various different *scale factors*, or *weights*, are applied at either event- or object-level. Event weights are for example used to normalize processes to a given integrated luminosity and to correct the MC pile-up profile to actually match the measured quantities in data. Weights applied at object-level take for example various object tagging efficiency and mistag rates into account and are applied in this analysis for the objects discussed in the following.

- **B-tagging weights** [166] are applied to scale the b-tagging efficiency and mistag rates to match those found in data, while additionally MC-to-MC scale factors are used to allow for the correction of different MC generators with different hadronization schemes. Fig. 9.11(a) shows the b-tagging scale factors together with their uncertainties as a function of the jet p_{T} for the $\epsilon_b = 77\%$ b-tagging working point.
- **C-tagging weights** were derived for this analysis and are applied in a similar fashion as b-tagging weights to account for data-MC or MC-MC efficiency and mistag rate differences in the c-tagging algorithm. Fig. 9.11(b) shows the c-tagging scale factors together with their preliminary uncertainties as a function of the jet p_{T} for the $\epsilon_c = 20\%$ c-tagging working point. As the full c-tagging scale factor uncertainties are not yet fully available at the time of writing, a conservative 30% estimate across the entire p_{T} range will be used instead of the preliminary uncertainties shown in the plot.
- **Top-tagging weights** [163] are again applied in a similar way as b- and c-tagging weights to correct the top-tagging efficiency and mistag rates.
- Further scaling factors are applied for example to jets based on the used JVT [159] working point to define signal jets and to leptons based on isolation, identification and trigger matching criteria.



(a) (b)

Figure 9.11: Scale factors and their uncertainties for (a) the `DL1r` b-tagging efficiency at the chosen $\epsilon_b = 77\%$ b-tagging working point [166] and (b) the `DL1r`$_c$ c-tagging efficiency at the chosen $\epsilon_c = 20\%$ c-tagging working point as function of the jet p_{T}. Note that the analysis makes use of a conservative 30% estimate on the c-tagging scale factor uncertainties instead of the (preliminary) c-tagging scale factor uncertainties shown in (b).

9.5 Main Discriminants

Based on the above discussed physics objects, this analysis defines different kinematic variables with good discrimination power between signal and background events that will be used for the definition of SRs, CRs and VRs.
In addition to basic kinematic variables, such as missing transverse momentum $E_{\mathrm{T}}^{\mathrm{miss}}$, jet transverse momenta p_{T} or also the number of jets, b-jets, c-jet and top-tagged jets (N_{jets}, $N_{\text{b-jets}}$, $N_{\text{c-jets}}$ and $N_{\text{top-jets}}$, respectively), more elaborated variables are defined and will be explained in the following.

Minimum Angular Distance between Jets and $E_{\mathrm{T}}^{\mathrm{miss}}$

The minimum angular distance in ϕ between the first n leading jets (i.e. the n jets with the highest p_{T}) and the $E_{\mathrm{T}}^{\mathrm{miss}}$ vector, denoted as $\min[\Delta\phi(\mathrm{jet}_{1-n}, E_{\mathrm{T}}^{\mathrm{miss}}]$, is defined to reduce the contamination from QCD events with highly mismeasured jet transverse momenta, which can cause high $E_{\mathrm{T}}^{\mathrm{miss}}$ values, even though no invisible particles are present. In these cases the $E_{\mathrm{T}}^{\mathrm{miss}}$ direction is usually aligned with the p_{T} of one of the leading jets and a lower cut on the minimum angular distance can therefore be very effective to reduce this contamination.

Transverse Mass

The *transverse mass* m_{T} can be used to constrain the mass of a particle that decays semi-invisible into a visible particle with transverse momentum p_{T} and one invisible particle giving rise to $E_{\mathrm{T}}^{\mathrm{miss}}$. In the approximation of two massless daughter particles, the transverse

mass is defined as [130]

$$m_{\mathrm{T}} = m_{\mathrm{T}}(p_{\mathrm{T}}, E_{\mathrm{T}}^{\mathrm{miss}}) = \sqrt{2 p_{\mathrm{T}} E_{\mathrm{T}}^{\mathrm{miss}} (1 - \cos(\Delta\phi))}, \tag{9.6}$$

where $\Delta\phi$ denotes the angles between the two daughter particles in the transverse plane. The distribution of m_{T} possesses an endpoint at the mass of the decaying mother particle. The transverse mass between the jet closest to $E_{\mathrm{T}}^{\mathrm{miss}}$ and $E_{\mathrm{T}}^{\mathrm{miss}}$, in the following referred to as $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$, has been found to considerably reduce e.g. backgrounds arising from $\mathrm{t\bar{t}}$ events, in which a tau from the leptonic W decay is reconstructed as a jet. The jet closest to $E_{\mathrm{T}}^{\mathrm{miss}}$ tends to be the misidentified tau, while the $E_{\mathrm{T}}^{\mathrm{miss}}$ arises from the undetected neutrino. In these cases, $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ is therefore expected to be bound by the W mass.

Additionally, the minimum and maximum transverse masses between all b-tagged jets and $E_{\mathrm{T}}^{\mathrm{miss}}$ as well as all c-tagged jets and $E_{\mathrm{T}}^{\mathrm{miss}}$, denoted as $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{f}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ and $\max[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{f}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ with flavor $\mathrm{f} = \mathrm{b}, \mathrm{c}$, are defined. While signal topologies will feature high values for these variables, background events like $\mathrm{t\bar{t}}$ will usually tend to much smaller values.

Effective Mass

The so-called *effective mass* m_{eff} is defined as

$$m_{\mathrm{eff}} = \sum_{j \in \mathrm{jets}} p_{\mathrm{T}}^{j} + E_{\mathrm{T}}^{\mathrm{miss}}, \tag{9.7}$$

and is particularly useful to select events that feature a combination of large jet transverse momenta and high $E_{\mathrm{T}}^{\mathrm{miss}}$.

Stransverse Mass

The *stransverse mass* m_{T2} [167, 168] can be used to constrain the mass of a pair-produced particle that subsequently decays in a semi-invisible fashion, as schematically illustrated in Fig. 9.12. Since two invisible particles contribute to the total missing transverse momentum, and their respective contributions are not known, it is not possible to separately construct two transverse masses for each decaying mother particle. If the individual contributions of the two invisible particles, $E_{\mathrm{T}}^{\mathrm{miss},1}$ and $E_{\mathrm{T}}^{\mathrm{miss},2}$, were known, it would be possible to easily constrain the mass of the pair-produced mother particle $m_{\tilde{t}_1}$ using the maximum of the two individual transverse masses as

$$m_{\tilde{t}_1}^2 \geq \max\left\{ m_{\mathrm{T}}^2(p_{\mathrm{T}}^{j1}, E_{\mathrm{T}}^{\mathrm{miss},1}), m_{\mathrm{T}}^2(p_{\mathrm{T}}^{j2}, E_{\mathrm{T}}^{\mathrm{miss},2}) \right\}, \tag{9.8}$$

where p_{T}^{j1} and p_{T}^{j2} denote the transverse momenta of the associated reconstructable objects. However, since these individual missing transverse momentum components are not known, the stransverse mass m_{T2} is defined as the minimum over all possible transverse momentum

combinations of Eq. (9.8) that lead to the total observed value of $E_{\mathrm{T}}^{\mathrm{miss}}$:

$$m_{\tilde{t}_1}^2 \geq m_{\mathrm{T2}}^2 = \min_{E_{\mathrm{T}}^{\mathrm{miss},1}+E_{\mathrm{T}}^{\mathrm{miss},2}=E_{\mathrm{T}}^{\mathrm{miss}}} \left[\max\left\{m_{\mathrm{T}}^2(p_{\mathrm{T}}^{j1}, E_{\mathrm{T}}^{\mathrm{miss},1}), m_{\mathrm{T}}^2(p_{\mathrm{T}}^{j2}, E_{\mathrm{T}}^{\mathrm{miss},2})\right\}\right] \tag{9.9}$$

As the presented analysis targets an asymmetric final state, in which one of the pair-produced stops decays into a top quark while the other decays into a charm quark, the m_{T2} variable will exploit this topology. The stop decay into a top quark will result in a large-radius jet that contains a small-radius b-tagged jet and is not expected to overlap with the charm-jet. The stransverse mass used in this analysis is therefore constructed with the p_{T} of the leading large-radius jet that contains a small-radius b-jet without any c-jet overlap $p_{\mathrm{T}}^{j;R=1.0;b;\not{c}}$ together with the p_{T} of the leading c-tagged jet p_{T}^{c-jet} :

$$m_{\mathrm{T2}}^2 = \min_{E_{\mathrm{T}}^{\mathrm{miss},1}+E_{\mathrm{T}}^{\mathrm{miss},2}=E_{\mathrm{T}}^{\mathrm{miss}}} \left[\max\left\{m_{\mathrm{T}}^2(p_{\mathrm{T}}^{j;R=1.0;b;\not{c}}, E_{\mathrm{T}}^{\mathrm{miss},1}), m_{\mathrm{T}}^2(p_{\mathrm{T}}^{c-jet}, E_{\mathrm{T}}^{\mathrm{miss},2})\right\}\right] \tag{9.10}$$

This definition will be particularly useful for signal points with large stop masses, as those will feature high endpoints in the m_{T2} distribution, while SM background processes such as $\mathrm{t\bar{t}}$ or single-top production will typically lead to much smaller values.

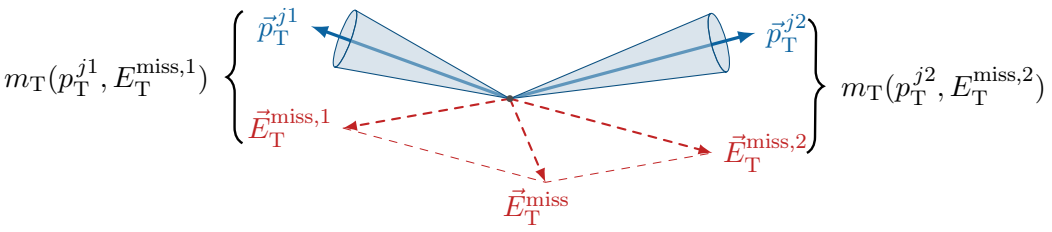


Figure 9.12: Schematic illustration of a pair-produced particle that undergoes a semi-invisible decay, resulting in two jets (illustrated in blue) and two invisible particles (red dashed lines). The mass of the mother particle can be constrained using the stransverse mass variable m_{T2}, which is based on the individual transverse mass components $m_{\mathrm{T}}(p_{\mathrm{T}}^{j1}, E_{\mathrm{T}}^{\mathrm{miss},1})$ and $m_{\mathrm{T}}(p_{\mathrm{T}}^{j2}, E_{\mathrm{T}}^{\mathrm{miss},2})$.

9.6 Analysis Strategy and Preselection

9.6.1 Analysis Strategy

The analysis strategy employed is largely dictated by the kinematics of the signal events and therefore by the mass difference $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$. Although all signal events are expected to contain at least one c-jet, one b-jet as well as high missing transverse momentum, the specific jet kinematics can largely vary depending on $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$: while large mass differences generally result in highly boosted and collimated objects, more separated jets with significantly different kinematic properties are expected for smaller $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$ close to the top mass. In order to ensure a high signal selection efficiency across all different considered kinematic regions, i.e. a variety of $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$ scenarios, the following three kinematic regions, as also schematically illustrated in Fig. 9.13, are defined:

- **Boosted Region (SRA)**: The boosted region, referred to as *SRA* in the following, targets final states with $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) \gg m_t$. Due to the large mass difference, signal events are expected to feature a high-p_{T} charm jet as well as a highly boosted top, whose collimated decay products are likely to be reconstructed in a large-radius jet with $R = 1.0$, as schematically illustrated in Fig. 9.14(a). The large mass difference additionally allows for a significant recoil of the system against the invisible neutralinos, resulting in high expected values of $E_{\mathrm{T}}^{\mathrm{miss}}$. A cut-and-count analysis based on dedicated discriminating variables, as discussed in Section 9.5, will be deployed in this region.

- **Intermediate Region (SRB)**: The intermediate region *SRB* is dedicated to signal kinematics with $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) > m_t$ and therefore targets slightly less boosted objects than SRA. While the decay topology of the top will be less collimated compared to SRA (as schematically shown in Fig. 9.14(b)), $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$ is still high enough to allow for a high energy transfer to the c-quark, resulting in an expected high-p_{T} charm jet in the final state. Unlike in SRA, the presence of a boosted top cannot be guaranteed any longer, which motivates the splitting of SRB into two separate regions: one still requiring at least one top-tagged large-radius jet (later referred to as region SRB1), while the other region targets the less boosted topologies and thus requires no top-tagged jets (later referred to as region SRB0). Furthermore, lower $E_{\mathrm{T}}^{\mathrm{miss}}$ values are expected while they should be still high enough to allow a rejection of most of the SM backgrounds. As for SRA, a cut-and-count analysis will be deployed.

- **Compressed Region (SRC)**: The compressed region *SRC* targets signal scenarios with $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) \approx m_t$, which lead to soft-$p_{\mathrm{T}}$ jets and low $E_{\mathrm{T}}^{\mathrm{miss}}$ values. In order to still guarantee a good discrimination power against SM background processes, the stop system is required to recoil against a high-p_{T} initial state radiation (*ISR*) jet, boosting the system towards higher jet transverse momenta and $E_{\mathrm{T}}^{\mathrm{miss}}$ values. The leading p_{T} jet will therefore be required to be a light jet with b- and c-veto. No top-tagged large-radius jet is expected in the compressed region, as the decay of the non-boosted top will result in well separated small-radius jets (see Fig. 9.14(c)). A dedicated neural network architecture is employed in this region to optimally separate signal from background events.

With the above described topologies, this analysis will define signal-enriched signal regions based on an $E_{\mathrm{T}}^{\mathrm{miss}}$-trigger and a 0L selection. The most dominant backgrounds are normalized in dedicated one-lepton (1L) and two-lepton (2L) control regions, whereas minor background processes are modeled using pure Monte Carlo methods. The background estimation methods are then verified in dedicated 0L validation regions, which are kinematically close, yet orthogonal, to the signal regions.

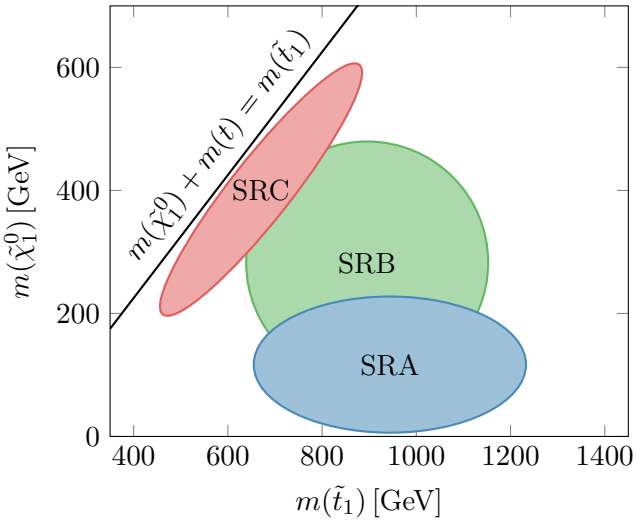
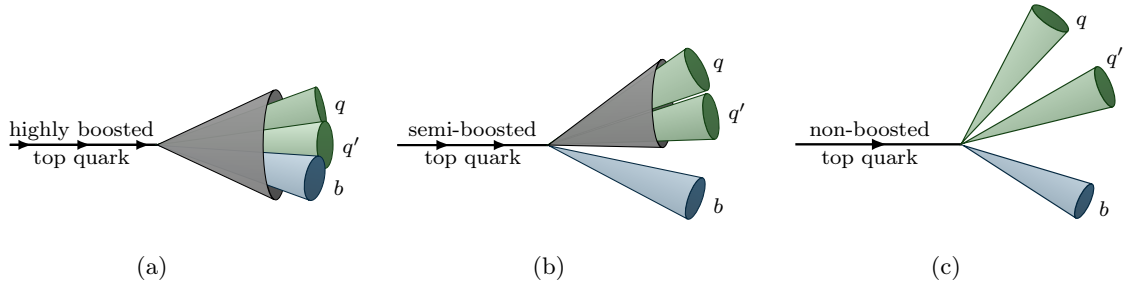


Figure 9.13: Schematic illustration of the tc + $E_{\mathrm{T}}^{\mathrm{miss}}$ analysis region strategy. While the boosted region SRA is dedicated to large mass differences $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) \gg m_t$, the compressed region SRC targets final states with $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) \approx m_t$. The intermediate region SRB is optimized for signal events with $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) > m_t$, which are not as boosted nor as compressed as in SRA and SRC, respectively.

9.6.2 SRA and SRB Preselection

Since SRA and SRB target similar signal topologies with large mass differences $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$, common preselections (referred to as SRAB preselection in the following) are defined for both region. The aim of these preselections is to reduce already at a first stage a large amount of contamination arising from $\mathrm{t\bar{t}}$ and multi-jet QCD backgrounds and thus to serve as a foundation for the following region optimization. While a 0L preselection is defined for signal and validation regions, 1L and 2L preselections will be used as a basis for the control region definitions. The SRAB preselection definitions are shown in Table 9.2.
All events are required to contain at least one reconstructed b- and c-jet as well as a minimum of three reconstructed jets for the 0L preselection, while only two jets are required in the 1L and 2L case in order to ensure enough statistics in the control regions. 0L and 1L preselection events are required to pass the $E_{\mathrm{T}}^{\mathrm{miss}}$ trigger and satisfy $E_{\mathrm{T}}^{\mathrm{miss}} > 250\,\mathrm{GeV}$ in order to be on the trigger plateau with good trigger efficiency. Additionally, a $\min[\Delta\phi(\mathrm{jet}_{1-4}, E_{\mathrm{T}}^{\mathrm{miss}})] > 0.4$ requirement is imposed to reduce multi-jet QCD backgrounds, while a $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})] > 150\,\mathrm{GeV}$ cut is defined to diminish the $\mathrm{t\bar{t}}$ contamination. The 2L preselection is specifically designed to select Z+jets events with two same-flavor opposite-sign signal leptons and an invariant mass of the dilepton system m_{ll} around the Z mass peak. Instead of $\mathrm{Z} \to \nu\nu$ events, this selection targets $\mathrm{Z} \to ll$ events with $l = e, \mu$, which have the same kinematics as the invisible Z decays but feature reconstructable leptons in the final state. As these events typically do not pass the $E_{\mathrm{T}}^{\mathrm{miss}}$ trigger, a single lepton trigger is used instead. Moreover, the invisible Z decay is mimicked by treating the reconstructed leptons as invisible when calculating the event's missing transverse momentum, resulting in the lepton-p_{T} corrected missing transverse momentum quantity



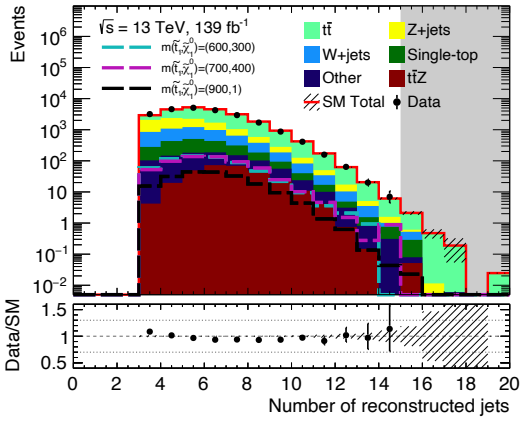
(a) (b) (c)

Figure 9.14: Schematic illustration of the expected full-hadronic top decay topology in (a) SRA, (b) SRB and (c) SRC. While the highly collimated decay products of the boosted top in (a) can be reconstructed as a large-radius jet (depicted in grey), the less boosted topology in (b) results in a reduced collimation and does therefore not always guarantee the presence of a reconstructable large-radius jet, motivating the splitting of SRB. The non-boosted top in (c) will result in well separated small-radius jets and no large-radius jet is expected.

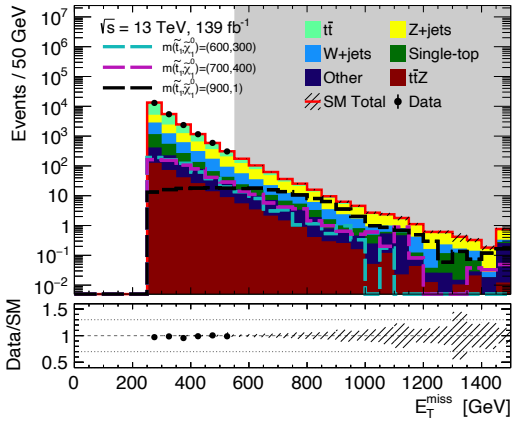
$E_{\mathrm{T},ll}^{\mathrm{miss}}$ with a requirement of $E_{\mathrm{T},ll}^{\mathrm{miss}} > 250\,\mathrm{GeV}$.
Various different observables at the common SRA and SRB 0L preselection level are shown in Fig. 9.15, where a good agreement between data and MC is observed. The main background contributions at the SRAB 0L preselection level are coming from $\mathrm{t\bar{t}}$ and Z+jets events, followed W+jets. For the sake of conciseness, 1L and 2L SRAB preselection plots are presented in Appendix B.2 and B.3, respectively.

Variable	**0L preselection**	**1L preselection**	**2L preselection**
Trigger	$E_{\mathrm{T}}^{\mathrm{miss}}$ Trigger		1L Trigger
$E_{\mathrm{T}}^{\mathrm{miss}}$ [GeV]	> 250		< 150
$E_{\mathrm{T},ll}^{\mathrm{miss}}$ [GeV]	–	–	> 250
Number of baseline leptons	0	–	–
Number of signal leptons	–	1	2
Dilepton system flavor	–	–	Same flavor
Dilepton system sign	–	–	Opposite sign
N_{jets}	≥ 3	≥ 2	
$N_{\text{b-jets}}$	≥ 1		
$N_{\text{c-jets}}$	≥ 1		
$\min[\Delta\phi(\mathrm{jet}_{1-4}, E_{\mathrm{T}}^{\mathrm{miss}})]$	> 0.4		–
m_{ll} [GeV]	–	–	[76,106]
Leading jet flavor	b- or c-tagged		
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	> 150		–

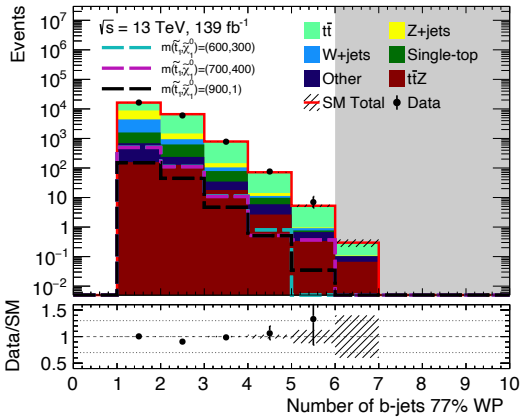
Table 9.2: Common preselections for regions SRA and SRB.



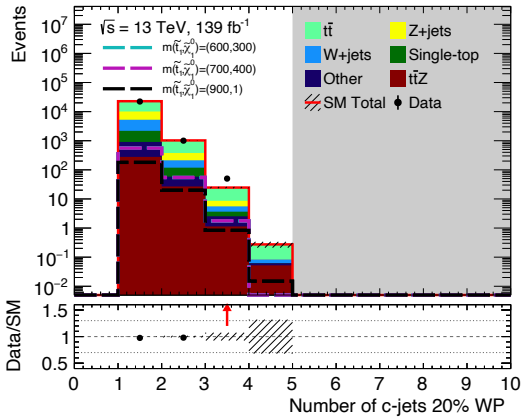
(a) N_{jets}



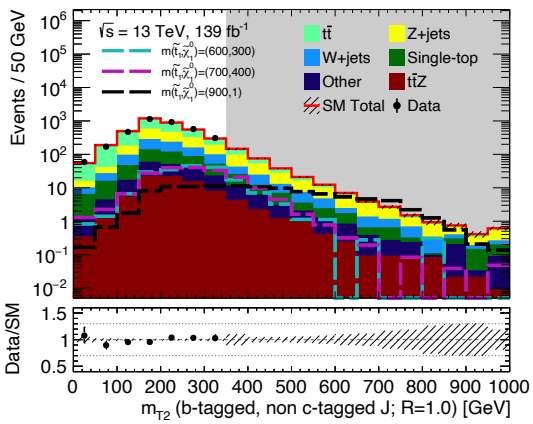
(b) $E_{\text{T}}^{\text{miss}}$



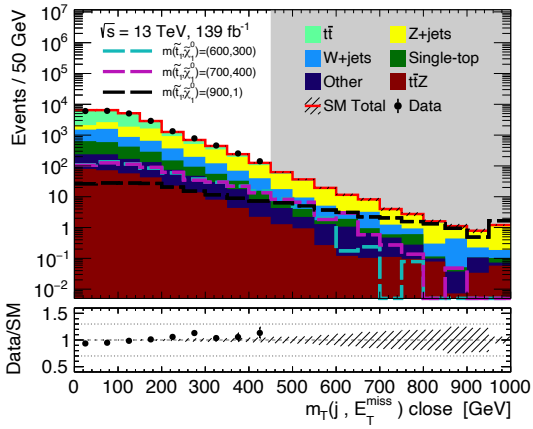
(c) $N_{\text{b-jets}}$



(d) $N_{\text{c-jets}}$



(e) m_{T2}



(f) $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$

Figure 9.15: Different observables shown at the common SRA and SRB 0L preselection level. Stacked SM background contributions are depicted together with expected signal yields for three different signal scenarios (dashed lines). Data is shown in areas where no significant excess in signal is expected, while signal-enhanced areas are blinded (grey band). A ratio between data and SM background predictions is shown in the bottom panel, where statistical uncertainties are indicated by shaded error bands. Signal point masses are given in GeV.

9.6.3 SRC Preselection

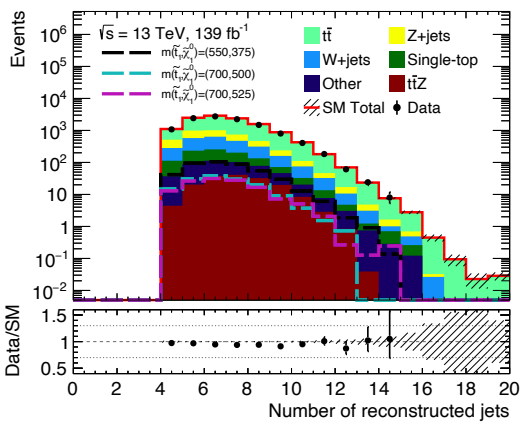
Since the signal topology in the compressed region SRC with small $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$ is significantly different from the topology expected in SRA and SRB, dedicated preselections are defined for SRC and its control and validation regions, again aiming to reduce $t\bar{t}$ and multi-jet QCD background contributions in the compressed phase space and to serve as a foundation for region optimization. A 0L preselection is defined as a basis for signal and validation region optimization, while 1L and 2L preselections will be used for control regions.

Table 9.3 summarizes the SRC preselection definitions. While basic selection criteria, such as the numbers of required jets or the applied triggers, are very similar to the SRAB preselections, the SRC preselection loosens the requirement of the minimum azimuthal angle between the first leading jets and the $E_{\mathrm{T}}^{\mathrm{miss}}$ to $\min[\Delta\phi(\mathrm{jet}_{1-3}, E_{\mathrm{T}}^{\mathrm{miss}})] > 0.3$. This selection, together with an additionally required $\mathcal{S} > 6$ cut, was found to reject more background contamination while allowing to pass more signal events in the compressed region. The stop system is required to recoil against a high-p_{T} ISR jet by imposing and b- and c-veto on the leading jet flavor. A lepton-p_{T} corrected missing transverse momentum quantity $E_{\mathrm{T},ll}^{\mathrm{miss}}$ is again used in the 2L preselection, while all $E_{\mathrm{T}}^{\mathrm{miss}}$-dependent variables use the corrected quantity $E_{\mathrm{T},ll}^{\mathrm{miss}}$ implicitly.

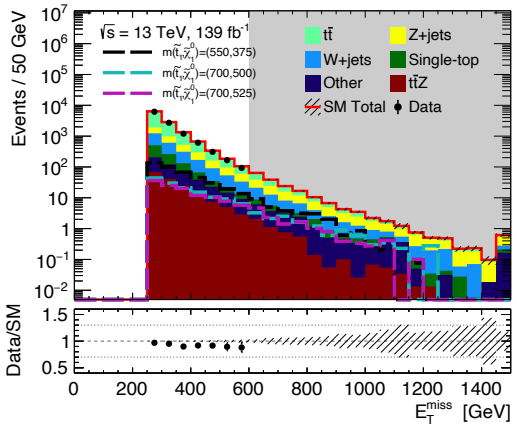
Various different observables at SRC 0L preselection level are shown in Fig. 9.16, where a good agreement between data and MC is observed. The majority of background events entering the SRC 0L preselection are $t\bar{t}$ events, followed by Z+jets and W+jets. For the sake of conciseness, 1L and 2L SRC preselection plots are presented in Appendix B.4 and B.5, respectively.

Variable	0L preselection	1L preselection	2L preselection
Trigger	$E_{\mathrm{T}}^{\mathrm{miss}}$ Trigger		1L Trigger
$E_{\mathrm{T}}^{\mathrm{miss}}$ [GeV]	> 250		< 100
$E_{\mathrm{T},ll}^{\mathrm{miss}}$ [GeV]	–	–	> 250
Number of baseline leptons	0	–	–
Number of signal leptons	–	1	2
Dilepton system flavor	–	–	Same flavor
Dilepton system sign	–	–	Opposite sign
N_{jets}	≥ 3		
$N_{\mathrm{b\text{-}jets}}$	≥ 1		
$N_{\mathrm{c\text{-}jets}}$	≥ 1		
$\min[\Delta\phi(\mathrm{jet}_{1-3}, E_{\mathrm{T}}^{\mathrm{miss}})]$	> 0.3		
m_{ll} [GeV]	–	–	[76,106]
Leading jet flavor	Light jet (b- and c-veto)		
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	> 100		
$\mathcal{S}$	> 6		
$p_{\mathrm{T}}^{\mathrm{jet}_1}$ [GeV]	> 100		
$p_{\mathrm{T}}^{\mathrm{jet}_4}$ [GeV]	> 30		
m_{T} [GeV]	–	> 30	–

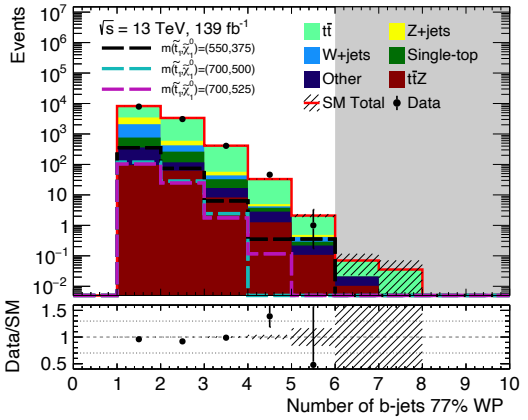
Table 9.3: Preselections for the region SRC.



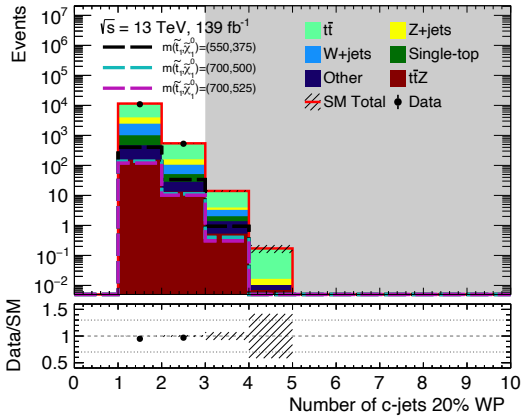
(a) N_{jets}



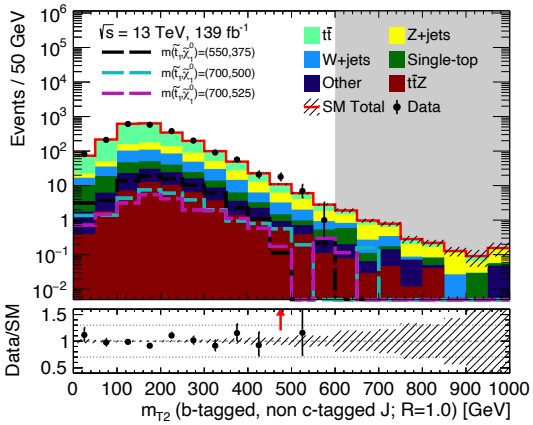
(b) $E_{\text{T}}^{\text{miss}}$



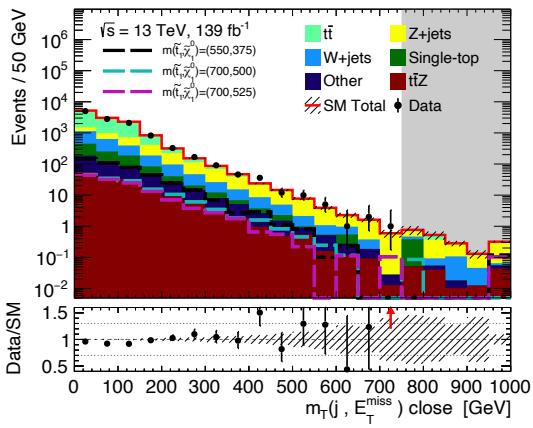
(c) $N_{\text{b-jets}}$



(d) $N_{\text{c-jets}}$



(e) m_{T2}



(f) $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$

Figure 9.16: Different observables shown at SRC 0L preselection level. Stacked SM background contributions are depicted together with expected signal yields for three different signal scenarios (dashed lines). Data is shown in areas where no significant excess in signal is expected, while signal-enhanced areas are blinded (grey band). A ratio between data and SM background predictions is shown in the bottom panel, where statistical uncertainties are indicated by shaded error bands. Signal point masses are given in GeV.

9.7 Signal Region Definitions

Based on the above defined preselections, dedicated signal regions are constructed to maximize the sensitivity to a variety of different signal scenarios. The following discusses the general optimization strategy employed in this analysis as well as the resulting definitions of the three signal regions SRA, SRB and SRC.

9.7.1 Region Optimization Strategy and N-1 Plots

The optimization of a signal region in this analysis starts with selecting a set of variables that are expected to well discriminate between the physics processes of background and signal events. The goal is then to find an optimal set of selection cuts on these variables that maximizes the signal discovery significance by reducing the background rates and enhancing the signal purity. Simulated MC background rates as well as their statistical uncertainties are taken into account and are compared to expected signal yields for representative benchmark signal points in the respective region to be optimized.
A commonly used approach in signal region optimization to find an optimal set of selection requirements is based on so-called *N-1 plots*. Here, one single discriminating variable is selected at a time and its distributions for the considered signal processes as well as the background components are plotted, applying the selection cuts on all N discriminants, except for the one being plotted (hence the name N-1). In addition to the signal and background distributions, a significance value is calculated at every position of the considered variable, under the assumption that a selection cut on this variable is performed at the respective value. This approach allows to easily understand the impact of a single selection requirement on the overall signal significance value. In practice, the method is performed iteratively on all considered variables until a combination of selection requirements is found that yields the best significance values.
However, since this method can become very time consuming, especially when dealing with a large number of discriminating variables, an automated *grid search* approach can be used. Here, discrete trial steps are defined for all considered variables and all different cut combinations of the resulting multidimensional grid are tested.
The presented analysis makes use of a combination of the above discussed approaches, where first initial grid searches on selected variables are performed, followed by dedicated fine-tunings with N-1 plots.
In order to allow for a statistical combination of the defined signal regions during the fitting procedure, all regions will be made orthogonal to one another through dedicated cuts, as explained in the respective sections.

9.7.2 Signal Region A

The aim of SRA is to target the bulk region of the considered signal mass plane, with mass differences of about $400\,\text{GeV} \lesssim \Delta m(\tilde{t}_1, \tilde{\chi}_1^0) \lesssim 1000\,\text{GeV}$, where highly boosted stop decay products are expected. The top quark from one of the stop decays is expected to be highly boosted and can therefore be reconstructed as a large-radius jet, while also the charm jet

from the other stop decay will exhibit very high transverse momentum. Therefore, the SRA selection, defined on top of the 0L SRAB preselection, will require at least one top-tagged large-radius jet, which already significantly reduces the contamination of V+jets background events, while the leading small-radius jet is required to be b- or c-tagged by the preselection. In order to reduce further background contaminations, specifically from $t\bar{t}$ and Z+jets events, different transverse mass variables with the most promising separation power between signal and background events as well as the above defined stransverse mass variable m_{T2} are selected for the following optimization procedure. Together with additional kinematic variables, such as the missing transverse energy significance $\mathcal{S}$ or the leading b- and c-jet transverse momenta $p_{\mathrm{T}}^{b_1}$ and $p_{\mathrm{T}}^{c_1}$, a multidimensional grid of discrete cut values is formed. Table 9.4 lists all selected observable as well as their respective cut values, for which an automated grid scan over all possible combinations is performed.
After an additional fine-tuning using an iterative N-1 plot approach, the SRA definition results in the set of selection requirements listed in Table 9.5. The N-1 plots of the main variables used to define SRA are shown in Fig. 9.17, where the impact of the chosen cuts on the signal and background rates can be observed. The requirement of at least one top-tagged large-radius jet as well as a high m_{T2} cut significantly reduce the Z+jets background. However, although no real boosted top jets are expected in Z+jets background events, a non-negligible number of these background events pass the imposed requirement. This behavior can be attributed to the very low background rejection power of the applied top-tagging algorithm (i.e. a large number of jets not originating from a real top will be misidentified as a top jet) as was shown in Fig, 9.10, which results from the high top-tagging efficiency required, as already the c-tagging algorithm is employed at only a 20% working point. Thus, even though a top jet requirement is imposed in SRA, the Z+jets background will be the leading background in this signal region. Furthermore, the N-1 plots show that the top background contributions can be significantly reduced by the transverse mass variables, while a large proportion of signal events can be retained.
Additional requirements on $\min[\Delta\phi(\mathrm{jet}_{1-4}, E_{\mathrm{T}}^{\mathrm{miss}})]$ and more low-level variables, such as $p_{\mathrm{T}}^{b_1}$ and $p_{\mathrm{T}}^{c_1}$ as used to define the multidimensional scan grid, did not help to further increase the signal significance and are therefore not applied.
As SRA is specifically designed to target large mass differences $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$, it is most sensitive to the illustrated $(m_{\tilde{t}_1}, m_{\tilde{\chi}_1^0}) = (800\,\mathrm{GeV}, 1\,\mathrm{GeV})$ signal benchmark point, whereas slightly reduced sensitivities are seen for smaller mass differences as well as larger stop masses due to the drastically decreasing stop production cross-section for high stop masses. The m_{T2} N-1 plot given in Fig. 9.17(f) shows that signal points from different kinematic regimes will result in very different behaviors in this observable. It is thus not possible to design a single signal region that achieves optimal sensitivity in the entire parameter space to be covered by SRA. Instead, SRA uses two orthogonal signal region bins in m_{T2} for the model-dependent exclusion fit (referred to as SRA-450 and SRA-575), while the model-independent discovery fit is inclusive with $m_{\mathrm{T2}} > 450\,\mathrm{GeV}$.
The expected pre-fit SM background yields as well as yields for selected SRA benchmark

signal points after SRA selection are shown in Table 9.6. Z+jets and single-top are the dominant backgrounds, for which dedicated control regions will be designed in Section 9.8.1.

Variable		Scanned cut values
$\mathcal{S}$	$>$	$\{0, 10, 15, 20, 25\}$
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	$>$	$\{0, 150, 200, 250, 300, 350, 400\}$
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	$>$	$\{150, 200, 250, 300, 350, 400\}$
$m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ [GeV]	$>$	$\{0, 50, 100, 150, 200, 250, 300\}$
m_{T2} [GeV]	$>$	$\{0, 300, 350, 400, 450, 500, 550, 600\}$
$\min[\Delta\phi(\mathrm{jet}_{1-4}, E_{\mathrm{T}}^{\mathrm{miss}})]$	$>$	$\{0.4, 0.6, 0.8, 1.0\}$
$p_{\mathrm{T}}^{c_1}$ [GeV]	$>$	$\{0, 100, 200, 300, 400\}$
$p_{\mathrm{T}}^{b_1}$ [GeV]	$>$	$\{0, 100, 200, 300, 400\}$

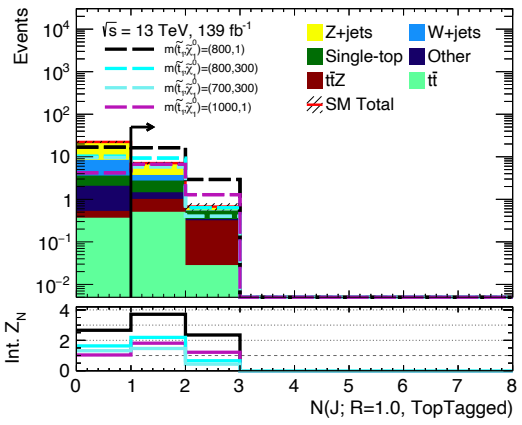
Table 9.4: Discriminating variables and their respective cut values used for the multidimensional grid scan for the SRA optimization. Cut values of 0 indicate that the optimization algorithm was not required to impose any cut on the respective variable.

Variable	SRA
Preselection	SRAB 0L preselection
$N_{\text{top-jets}}$	≥ 1
$\mathcal{S}$	> 18
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	> 200
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	> 200
$m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ [GeV]	> 100
m_{T2} [GeV]	$[450, 575)$, $[575, \infty)$

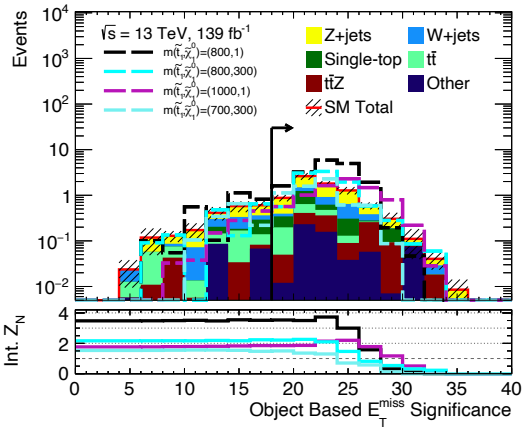
Table 9.5: SRA definition on top of the 0L SRAB preselection.

Process	SRA
Z+jets	3.51 ± 0.24
Single-top	1.51 ± 0.24
W+jets	0.97 ± 0.23
$t\bar{t}Z$	0.81 ± 0.14
Other	0.56 ± 0.18
$t\bar{t}$	0.52 ± 0.14
SM	7.88 ± 0.46
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (800, 1)$ GeV	17.88 ± 1.18
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (800, 300)$ GeV	9.95 ± 0.90
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (1000, 1)$ GeV	8.03 ± 0.53
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (700, 300)$ GeV	6.30 ± 0.75

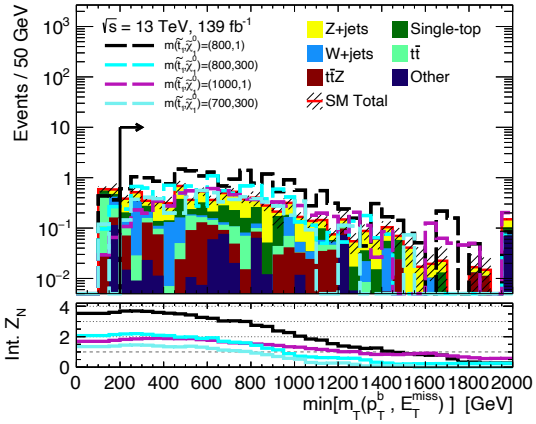
Table 9.6: Expected pre-fit SM background and selected signal benchmark point yields for SRA selection. Only statistical uncertainties are shown.



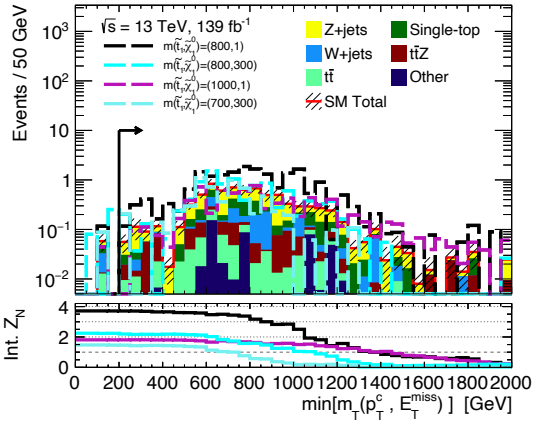
(a) $N_{\text{top-jets}}$



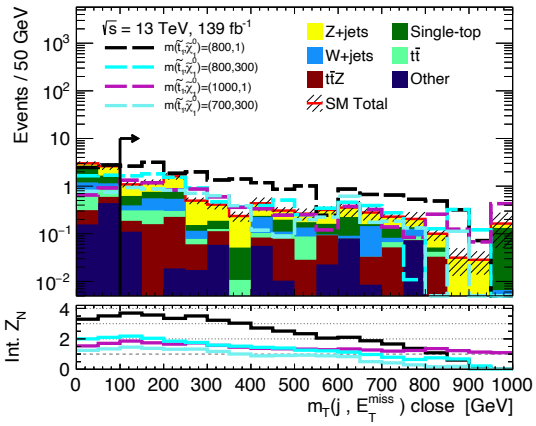
(b) $\mathcal{S}$



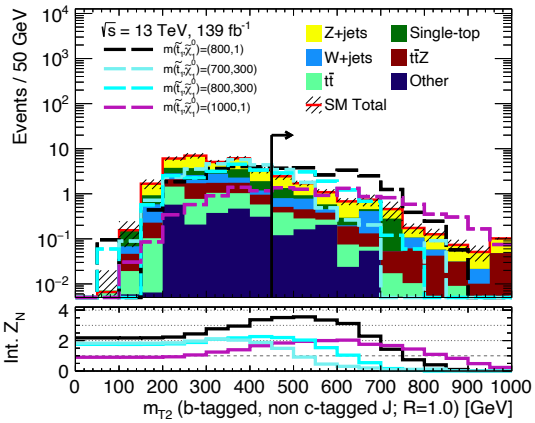
(c) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$



(d) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$



(e) $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$



(f) m_{T2}

Figure 9.17: N-1 plots of the main discriminating variables used in the SRA definition, showing background and selected SRA benchmark signal points. The black arrow indicates the chosen cut value of the respective variable and the bottom panel shows the significance value Z_N when integrating over all events after the applied cut in the direction of the displayed arrow. Signal point masses are given in GeV and only statistical uncertainties are shown.

9.7.3 Signal Region B

SRA already provides a high sensitivity to a large area in the $(m_{\tilde{t}_1}, m_{\tilde{\chi}_1^0})$ parameter space for both large and also more intermediate mass differences. However, as SRA is designed to specifically target highly boosted topologies, it is less sensitive to smaller mass differences of around $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) \simeq 300\,\text{GeV}$. The intermediate signal region SRB is therefore defined to complement SRA in this region of parameter space.

As already schematically illustrated in Fig. 9.14, the less boosted top-decay topology will not any longer guarantee the presence of highly collimated decay products and thus, a top-tagged large-radius jet might not be reconstructed, while signal events are additionally expected to present lower values for multiple kinematic variables, such as m_{T2} or missing transverse momentum. SRB is therefore split into two separate regions: events with no top-tagged large-radius jet are targeted by SRB0, whereas SRB1 is dedicated to signal events containing at least one reconstructed top-tagged large-radius jets but with lower m_{T2} than SRA. The definitions of SRB0 and SRB1, as again obtained from a combined parameter scan and N-1 plot approach, is presented in Table 9.7.

Since the dropped top-tagged large-radius jet requirement in SRB0 significantly increases the contamination from V+jets events, additional cuts on multiple transverse mass as well as on leading jet p_{T} variables are imposed. Compared to SRA, the m_{T2} or missing transverse momentum significance $\mathcal{S}$ is considerably lower, in order to account for the expected less boosted topologies. SRB1, on the other hand, retains the top-tagged large-radius jet requirement, targets however signal events that present smaller stransverse masses m_{T2}. While SRB0 is by construction orthogonal to SRA and SRB1 due to the top-tagging requirement, SRB1 is made orthogonal to SRA by the kinematic m_{T2} cut. N-1 plots for both SRB0 and SRB1 are shown in Fig. 9.18 and Fig. 9.19, respectively, where again the influence of the respective selection requirements on the various background suppressions can be observed.

Furthermore, the $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$ variable is used to define bins in both SRB0 and SRB1, labeled according to their respective lower $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$ boundaries, such as e.g. SRB0-100 or SRB1-500, for the model-dependent exclusion fit, while both regions will be inclusive in this variable with $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}} > 100\,\text{GeV}$ for the model-independent discovery fit.

The expected SRB0 and SRB1 pre-fit SM background and signal yields are shown in Table 9.8, where Z+jets is identified as the dominant background in both regions. While this is expected in case of SRB0, where no top-tagged jet is required, the large rate in SRB1 again results from the significant mistag rate of the employed top-tagging algorithm, similar to what was observed in SRA. Subleading backgrounds are W+jets for SRB0 and single-top for SRB1. Dedicated control region definitions for SRB will be discussed in Section 9.8.1.

Variable	SRB0	SRB1
Preselection	SRAB 0L preselection	
$N_{\text{top-jets}}$	0	≥ 1
$E_{\text{T}}^{\text{miss}}$ [GeV]	> 300	—
N_{jets}	≥ 5	≥ 3
$p_{\text{T}}^{b_1}$ [GeV]	–	> 100
$p_{\text{T}}^{c_1}$ [GeV]	>100	
$p_{\text{T}}^{c_2}$ [GeV]	> 100	–
$p_{\text{T}}^{j_4}$ [GeV]	> 50	–
$\min[m_{\text{T}}(p_{\text{T}}^{\text{b}}, E_{\text{T}}^{\text{miss}})]$ [GeV]	> 200	
$\max[m_{\text{T}}(p_{\text{T}}^{\text{b}}, E_{\text{T}}^{\text{miss}})]$ [GeV]	[200,700]	–
$\min[m_{\text{T}}(p_{\text{T}}^{\text{c}}, E_{\text{T}}^{\text{miss}})]$ [GeV]	> 150	> 300
$\max[m_{\text{T}}(p_{\text{T}}^{\text{c}}, E_{\text{T}}^{\text{miss}})]$ [GeV]	> 400	–
$\mathcal{S}$	> 10	> 17
m_{T2} [GeV]	> 150	[200,450]
$m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$ [GeV]	[100,150), [150,400), $[400, \infty)$	[100,150), [150,300), [300,500), $[500, \infty)$

Table 9.7: SRB0 and SRB1 definitions on top of the 0L SRAB preselection.

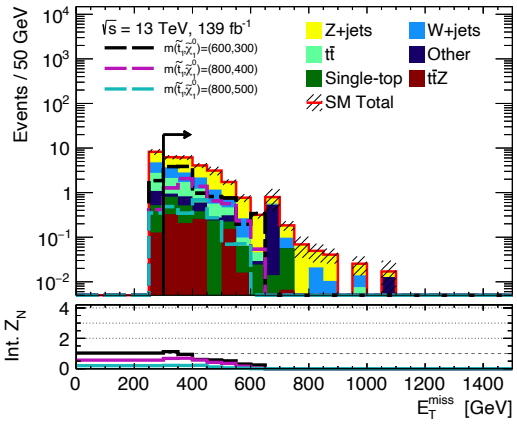
Process	SRB0	SRB1
Z+jets	11.52 ± 1.12	8.21 ± 0.82
W+jets	4.46 ± 0.69	2.53 ± 0.51
Single-top	1.32 ± 0.24	3.13 ± 0.45
$t\bar{t}$	3.00 ± 0.25	1.73 ± 0.22
$t\bar{t}$Z	0.90 ± 0.16	1.93 ± 0.20
Other	1.32 ± 0.46	1.56 ± 0.42
SM	22.52 ± 1.45	19.10 ± 1.18
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (600, 300)$ GeV	16.79 ± 1.54	10.40 ± 1.19
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (800, 400)$ GeV	7.91 ± 0.78	6.39 ± 0.74
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (800, 500)$ GeV	3.36 ± 0.51	2.31 ± 0.41

Table 9.8: Expected pre-fit SM background and selected signal benchmark point yields for SRB0 and SRB1 selection. Only statistical uncertainties are shown.

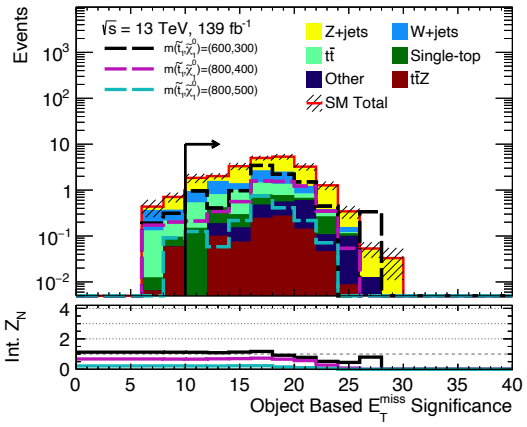
9.7.4 Signal Region C

The discrimination between signal and background events for the compressed SRC topology with $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) \approx m_t$ is largely based on a dedicated multi-class classification neural network, which uses low-level input variables, such as p_{T}, η, $\Delta\phi(\text{jet}, E_{\text{T}}^{\text{miss}})$ and flavor of the six leading jets, $E_{\text{T}}^{\text{miss}}$ as well as N_{jets}, $N_{\text{b-jets}}$, $N_{\text{c-jets}}$ and $N_{\text{top-jets}}$ variables, to produce three event classification scores: a signal score NN_{score}, a V+jets score $\text{NN}_{\text{score}}^{\text{V+jets}}$ and a $\text{t}\bar{\text{t}}$ score $\text{NN}_{\text{score}}^{\text{t}\bar{\text{t}}}$. The neural network was trained on background events as well as signal samples with $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) = 175\,\text{GeV}$ and $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0) = 200\,\text{GeV}$.

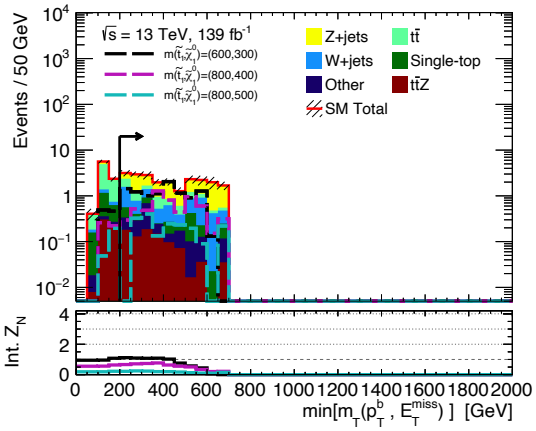
Signal region C is defined on top of the 0L SRC preselection to enforce an ISR topology, while additional selections on the distance between the two leading jets, $\Delta R(j_1, j_2)$, as well



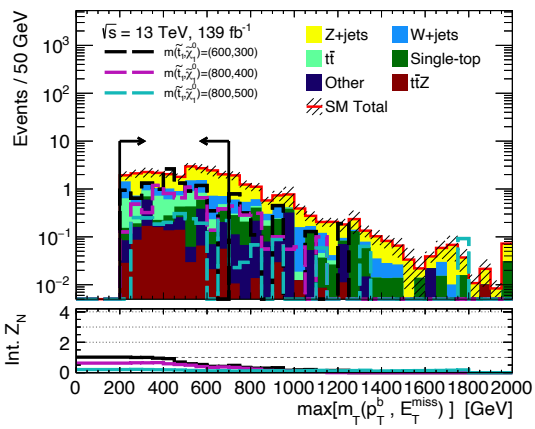
(a) $E_{\mathrm{T}}^{\mathrm{miss}}$



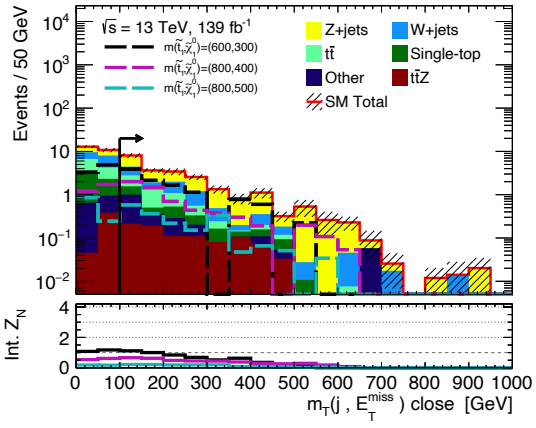
(b) $\mathcal{S}$



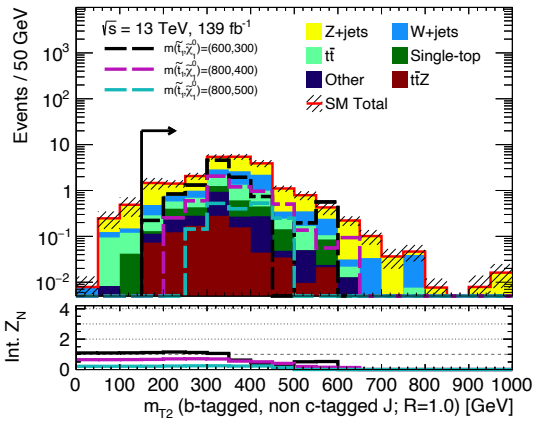
(c) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$



(d) $\max[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$

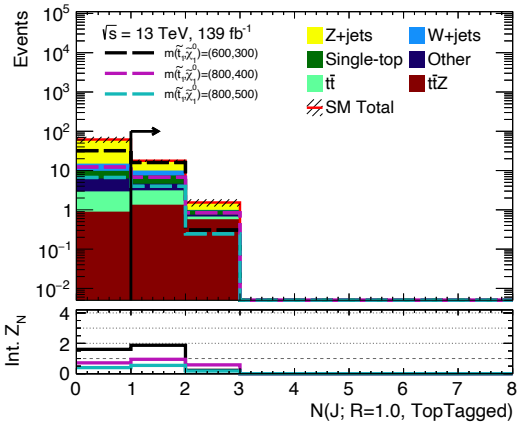


(e) $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$

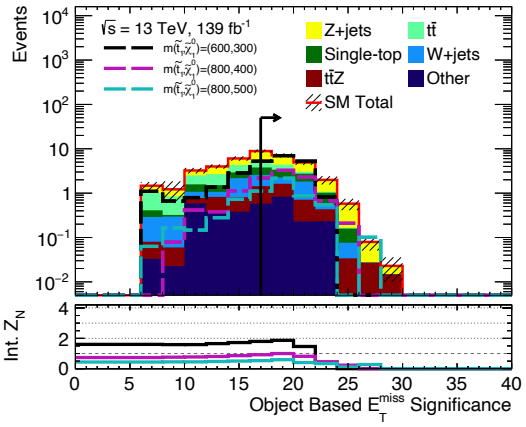


(f) m_{T2}

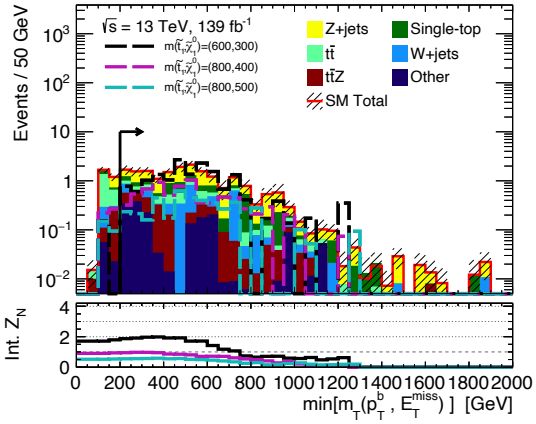
Figure 9.18: N-1 plots of selected discriminating variables used in the SRB0 definition, showing background and selected SRB benchmark signal points. The black arrow indicates the chosen cut value of the respective variable and the bottom panel shows the significance value Z_N when integrating over all events after the applied cut in the direction of the displayed arrow. Signal point masses are given in GeV and only statistical uncertainties are shown.



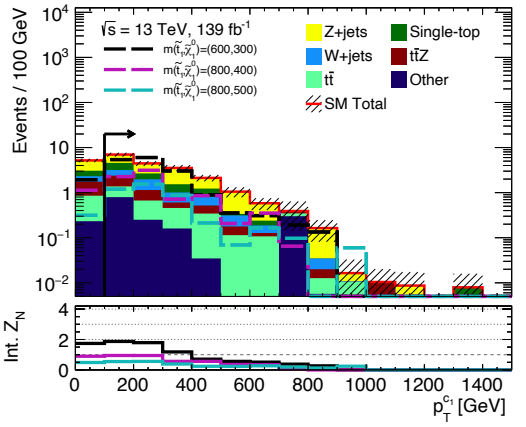
(a) $N_{\text{top-jets}}$



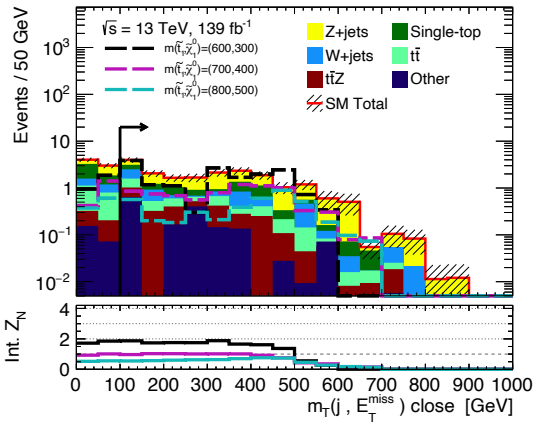
(b) $\mathcal{S}$



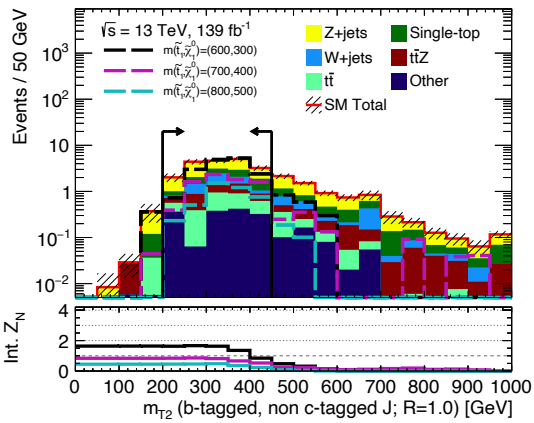
(c) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$



(d) $p_{\mathrm{T}}^{\mathrm{c1}}$



(e) $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$



(f) m_{T2}

Figure 9.19: N-1 plots of selected discriminating variables used in the SRB1 definition, showing background and selected SRB benchmark signal points. The black arrow indicates the chosen cut value of the respective variable and the bottom panel shows the significance value Z_N when integrating over all events after the applied cut in the direction of the displayed arrow. Signal point masses are given in GeV and only statistical uncertainties are shown.

as on the signal score $\mathrm{NN}_{\mathrm{score}}$ and m_{eff} are used to suppress the dominant $\mathrm{t\bar{t}}$ background. The SRC base definition is shown in Table 9.9, while additional requirements and binnings are applied on top of this selection for the model-dependent and model-independent fits, as discussed below. The N-1 plots of the main variables used to define SRC are shown in Fig. 9.20, where the impact of the chosen cuts on the expected signal and background rates can be observed.

The relatively loose SRC definition requirements guarantee a high signal selection efficiency, and therefore high signal statistics, enabling the use of a two-dimensional binning of SRC in m_{eff} and $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ for the model-dependent fit in order to target different signal scenarios and thus maximize the expected exclusion significance. The definitions of the binned SRC sub-regions for the model-dependent fit are given in Table 9.10, where each region is named after the lower m_{eff} cut value of the respective bin. The model-independent discovery fit, on the other hand, is inclusive in $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ with $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}} > 200\,\mathrm{GeV}$ but makes use of the same m_{eff} binning.

While all SRC sub-regions are by definition statistically independent to one another due to the orthogonality of the kinematic m_{eff} and $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ bins, all SRC regions are orthogonal to both the SRA and SRB regions by the leading jet flavor requirement. All regions can therefore be used simultaneously in the fit discussed in Section 9.10.

Table 9.11 lists the pre-fit expected SM background yields as well as the signal yields for selected SRC signal benchmark points for three example SRC sub-regions. The dominant background process in all SRC regions is $\mathrm{t\bar{t}}$, followed by V+jets processes. The definition of dedicated control regions will be discussed in Section 9.8.

Variable	**SRC**
Preselection	SRC 0L preselection
$\mathrm{NN}_{\mathrm{score}}$	> 0.75
$\Delta R(j_1, j_2)$	> 1.0
m_{eff} [GeV]	> 750

Table 9.9: Definition of the SRC base selection on top of the 0L SRC preselection.

Region	$\mathbf{m_{eff}}$ **[GeV]**	$\mathbf{m_T(p_T^j, E_T^{miss})_{close}}$ **[GeV]**
SRC-750	[750,1000)	[0,100), [100,200), [200,300), [300,∞)
SRC-1000	[1000,1250)	[0,100), [100,200), [200,∞)
SRC-1250	[1250,1500)	[0,100), [100,∞)
SRC-1500	[1500,1750)	[0,100), [100,∞)
SRC-1750	[1750,2000)	–
SRC-2000	$\geq$ 2000	–

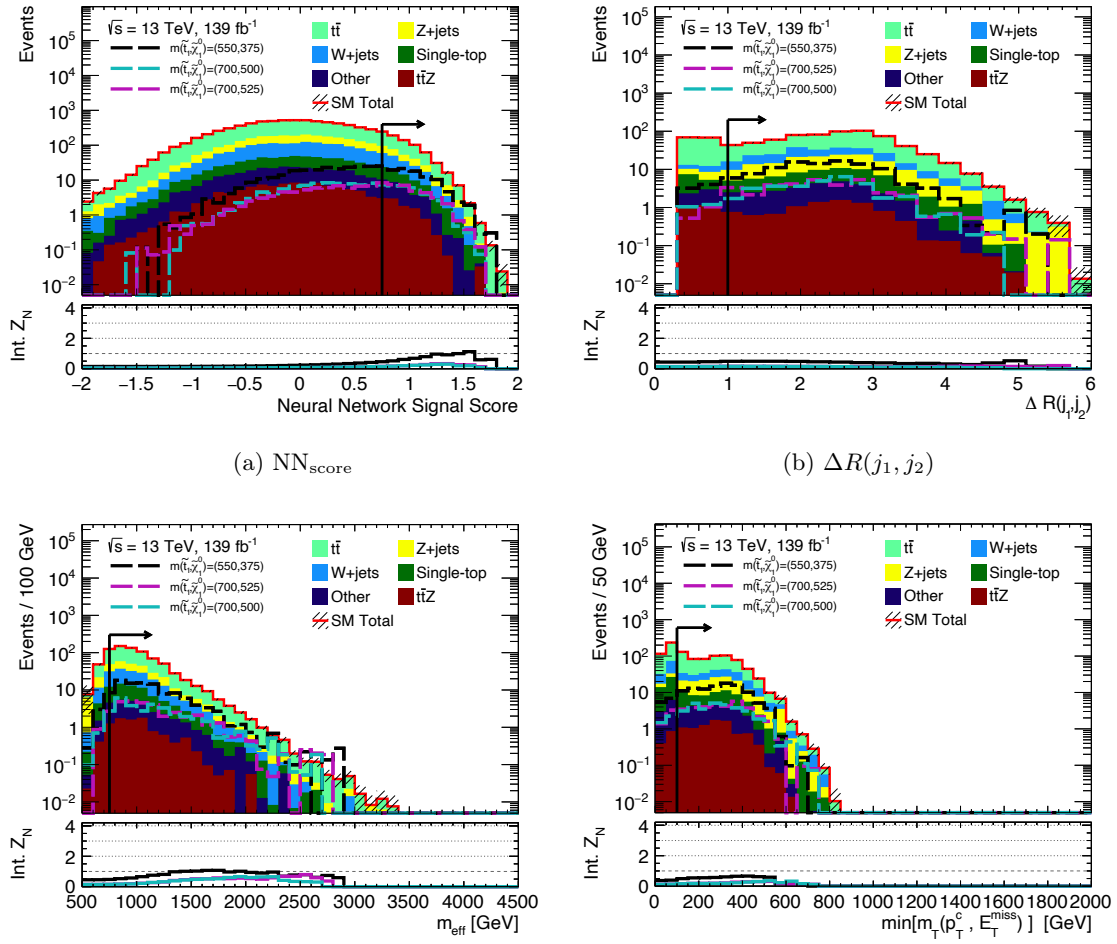
Table 9.10: SRC sub-region definitions in a two-dimensional binning in m_{eff} and $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$ on top of the SRC base selection for the model-dependent exclusion fit.

Process	**SRC-750**	**SRC-1500**	**SRC-2000**
$t\bar{t}$	220.00 ± 2.82	14.39 ± 0.45	2.82 ± 0.21
W+jets events	52.03 ± 2.71	3.49 ± 0.44	0.53 ± 0.08
Z+jets events	49.97 ± 2.60	3.77 ± 0.52	0.47 ± 0.06
Single-top	21.02 ± 1.29	2.09 ± 0.29	0.30 ± 0.10
Other	9.21 ± 0.93	1.33 ± 0.27	0.17 ± 0.07
$t\bar{t}$Z	4.25 ± 0.30	0.35 ± 0.09	0.02 ± 0.03
SM	356.48 ± 4.97	25.43 ± 0.91	4.31 ± 0.26
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (550, 375)$ GeV	37.87 ± 2.28	9.07 ± 1.14	2.55 ± 0.61
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (700, 500)$ GeV	11.87 ± 1.11	3.41 ± 0.60	1.58 ± 0.38
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (700, 525)$ GeV	11.85 ± 1.07	4.15 ± 0.61	1.38 ± 0.37

Table 9.11: Expected pre-fit SM background and selected signal benchmark point yields for the inclusive SRC-750, SRC-1500 and SRC-2000 sub-regions. Only statistical uncertainties are shown.

9.8 Estimation of Standard Model Backgrounds

As discussed above, the most dominant pre-fit Standard Model backgrounds entering both SRA and SRB are Z+jets events, followed by single-top, whereas $t\bar{t}$ and V+jets are the dominant backgrounds in SRC. This section describes the SM background estimation methods based on CRs to derive normalization corrections for these dominant MC predictions in all SRs as well as the validation of the employed estimation methods in dedicated VRs. All other background processes only have a negligible impact on the overall SM background and are therefore directly taken from their MC predictions.



(a) $\mathrm{NN}_{\mathrm{score}}$

(b) $\Delta R(j_1, j_2)$

(c) m_{eff}

(d) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$

Figure 9.20: N-1 plots of selected discriminating variables used in the SRC definition and 0L SRC preselection definition, showing background and selected SRC benchmark signal points. The black arrow indicates the chosen cut value of the respective variable and the bottom panel shows the significance value Z_N when integrating over all events after the applied cut in the direction of the displayed arrow. Signal point masses are given in GeV and only statistical uncertainties are shown.

9.8.1 Background Estimation in Signal Regions A and B

The signal topologies targeted by SRA and SRB are very similar and already motivated the use of a set of common preselections. The resulting dominant background processes entering these regions are therefore also very similar and hence allow the definition of common control regions for the dominant backgrounds in SRA and SRB. The following introduces the SRA and SRB control and validation region definitions which are based on the common 1L and 2L SRAB preselection definitions, as presented in Table 9.2.

Z Control Regions: CRAB1Z and CRB0Z

As the signal regions SRA and SRB1 present very similar kinematic selections, with SRB1 only targeting slightly less boosted topologies with a lower m_{T2} requirement and some additional jet kinematic cuts, a common Z+jets control region, called CRAB1Z, is designed to simultaneously fit the Z+jets background in the two regions. A separate Z+jets control region is furthermore designed for SRB0, referred to as CRB0Z. Both CRAB1Z and CRB0Z are used to define one common Z+jets scaling factor $\mu_{\mathrm{Z,AB}}$ for the Z+jets background in all SRA and SRB regions.

CRAB1Z is defined on top of the common SRAB 2L preselection, already guaranteeing a pure selection of Z+jets events, with an additional transverse momentum requirement imposed on the two leading leptons with $p_{\mathrm{T}}^{l_2} > 30\,\mathrm{GeV}$ to ensure a constant offline lepton trigger efficiency in the selected phase space, as shown in Table 9.12. In order to avoid extrapolation from the control region to the signal regions on the number of reconstructed jets (inclusive, b-jets and c-jets), the number of top-tagged large-radius jets and the $E_{\mathrm{T}}^{\mathrm{miss}}$ significance $\mathcal{S}$, CRAB1Z imposes the same selection criteria on these variables as its corresponding signal regions. Since the requirement of at least one top-tagged large-radius jet in this region drastically reduces the available statistics of Z+jets events, several transverse mass requirements are lifted or entirely removed. SRA and SRB1 were made orthogonal to each other by defining them as two orthogonal bins in m_{T2}, motivating the inclusiveness in m_{T2} in their common Z+jets control region without imposing any requirements on this variable. Note the implicit use of the lepton-p_{T} corrected $E_{\mathrm{T},ll}^{\mathrm{miss}}$ quantity for all $E_{\mathrm{T}}^{\mathrm{miss}}$-related calculations in this Z+jets CR section, unless stated otherwise.

Similarly, the definition of CRB0Z is also based on the common SRAB 2L preselection with the same additional two leading lepton transverse momentum requirement as imposed in CRAB1Z, while also an extrapolation on the number of jets, number of top-tagged large-radius jets as well as the $E_{\mathrm{T}}^{\mathrm{miss}}$ significance $\mathcal{S}$ is avoided by choosing the same selection as in SRB0. Most notably, CRB0Z requires exactly zero top-tagged large-radius jets, guaranteeing enough Z+jets statistics in this region. The same set of selection requirements on all transverse mass variables as used to define signal region SRB0 can therefore be imposed, making CRB0Z kinematically very close to its corresponding signal region.

Fig. 9.21 and 9.22 show selected N-1 plots and distributions of some key variables in CRAB1Z and CRB0Z, respectively.

A high purity in Z+jets events with only little contamination from other SM backgrounds is observed in the presented control regions. Table 9.13 summarizes these background yields as well as the purity of each contribution in CRAB1Z and CRB0Z and compares them to data observed in the control regions. Good data-MC agreement is seen for CRB0Z while the MC background prediction leads to a slight overestimation compared to data. Furthermore, none of the SRA and SRB signal benchmark events enter the defined control regions.

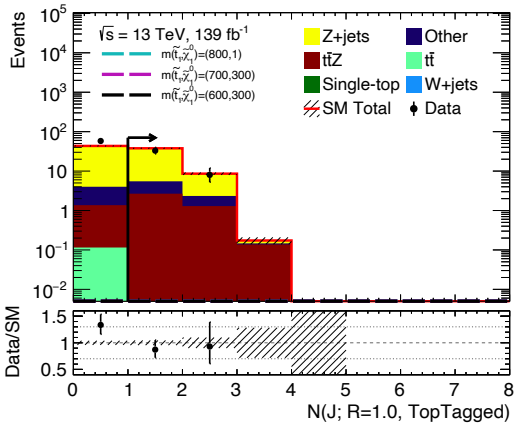
A dedicated validation region to validate the employed Z+jets background control region strategy for SRA and SRB will be presented below after the definition of the single-top control region for SRA and SRB1.

Variable	CRAB1Z	CRB0Z
Preselection	SRAB 2L preselection	
$p_{\mathrm{T}}^{l_2}$ [GeV]	> 30	
$N_{\text{top-jets}}$	≥ 1	0
N_{jets}	≥ 3	≥ 5
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	–	> 150
$\max[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	–	> 400
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	–	> 200
$\max[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	–	> 200
$m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\text{close}}$ [GeV]	–	> 100
$\mathcal{S}$	> 17	> 10
m_{T2}	–	

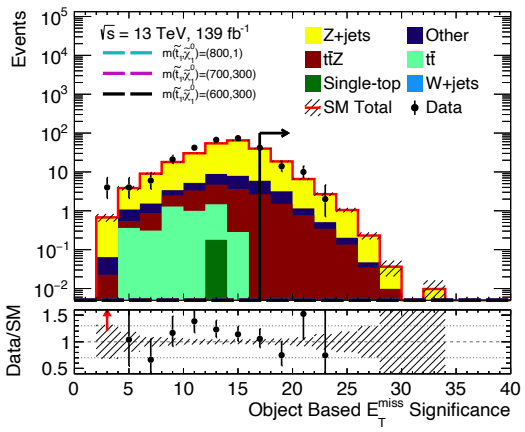
Table 9.12: Definition of CRAB1Z and CRB0Z on top of the common SRAB 2L preselection as defined in Table 9.2. Note that all $E_{\mathrm{T}}^{\mathrm{miss}}$-related quantities implicitly use the missing transverse energy of the event corrected by treating the two reconstructed leptons as invisible.

Process	CRAB1Z	CRB0Z
Z+jets	38.65 ± 1.25 (80.9%)	30.53 ± 1.18 (85.8%)
Other	5.02 ± 0.44 (10.5%)	3.19 ± 0.33 (9%)
$\mathrm{t\bar{t}Z}$	4.11 ± 0.16 (8.6%)	1.49 ± 0.10(4.2%)
$\mathrm{t\bar{t}}$	0.00 ± 0.00	0.27 ± 0.19 (0.8%)
Single-top	0.00 ± 0.00	0.09 ± 0.09(0.2%)
W+jets	0.00 ± 0.00	0.00 ± 0.00
SM	47.78 ± 1.33	35.58 ± 1.25
Data	41.00 ± 6.40	35.00 ± 5.92
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (800, 400)$ GeV	0.00 ± 0.00	0.00 ± 0.00
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (800, 1)$ GeV	0.00 ± 0.00	0.00 ± 0.00
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (700, 300)$ GeV	0.00 ± 0.00	0.00 ± 0.00
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (600, 300)$ GeV	0.00 ± 0.00	0.00 ± 0.00

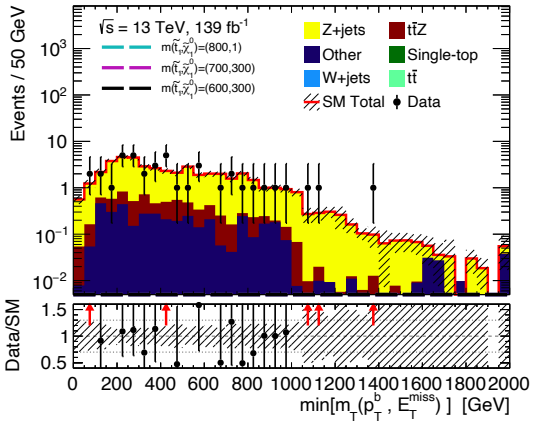
Table 9.13: Expected pre-fit SM background yields as well as data observed in CRAB1Z and CRB0Z. Respective relative SM contributions to the total CR yields are illustrated in brackets and only statistical errors are shown.



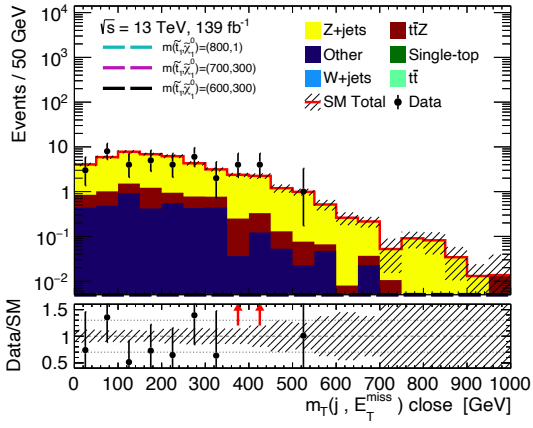
(a) $N_{\text{top-jets}}$



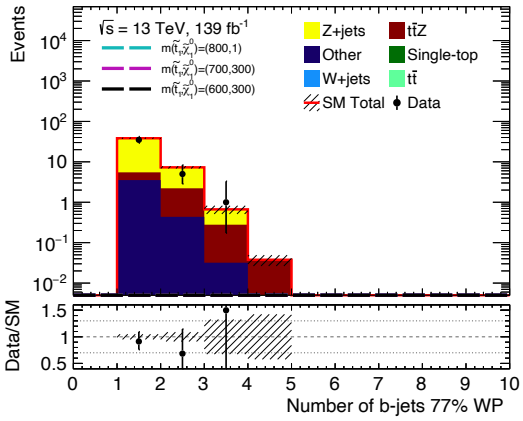
(b) $\mathcal{S}$



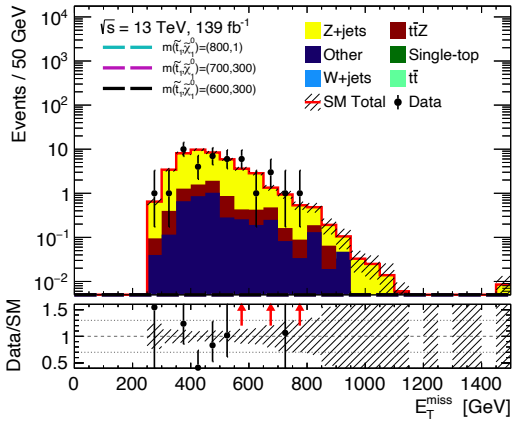
(c) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$



(d) $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$

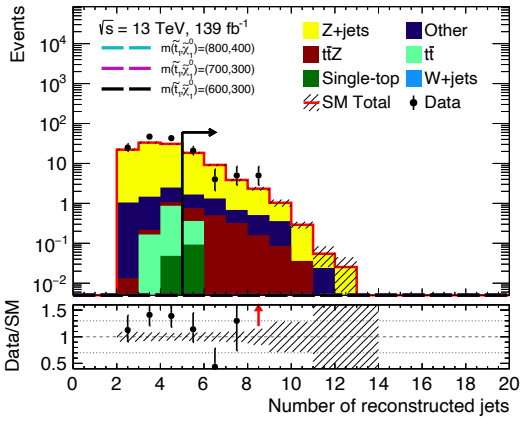


(e) $N_{\text{b-jets}}$

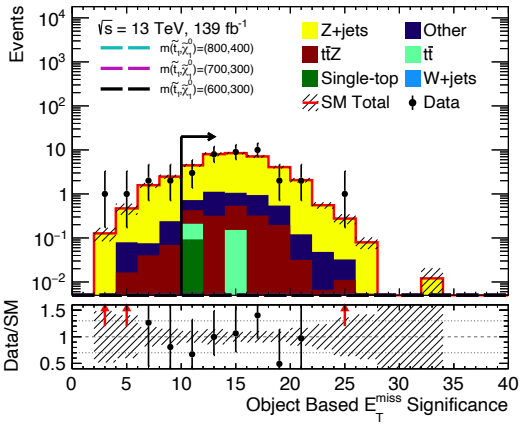


(f) $E_{\mathrm{T},ll}^{\mathrm{miss}}$

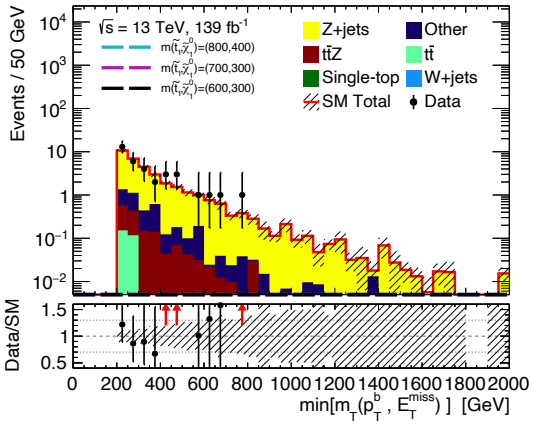
Figure 9.21: Selected N-1 plots for the CRAB1Z definition in (a) and (b) as well as distributions of some key variables in CRAB1Z in (c)-(f), showing background and selected SRA and SRB signal benchmark points. A high purity in Z+jets events with only little contamination from other background sources is observed, while no signal contamination in the presented CR is visible. Signal point masses are given in GeV and only statistical uncertainties are shown.



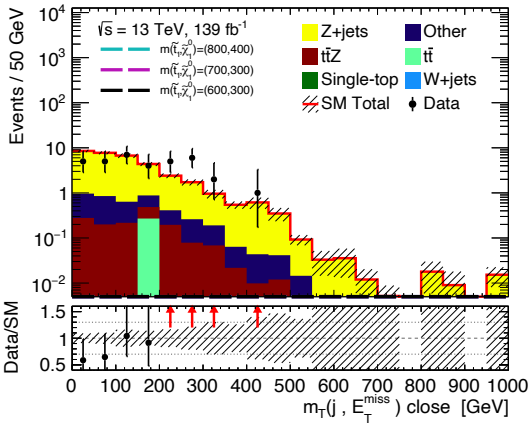
(a) N_{jets}



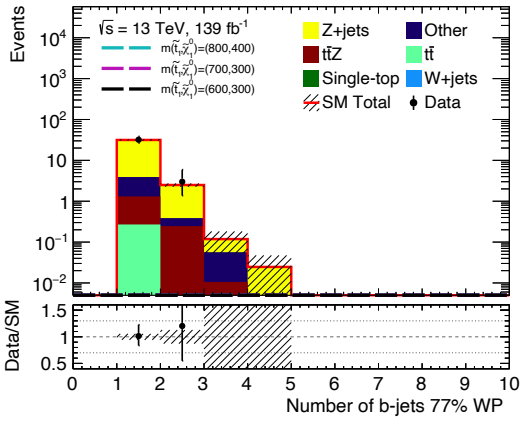
(b) $\mathcal{S}$



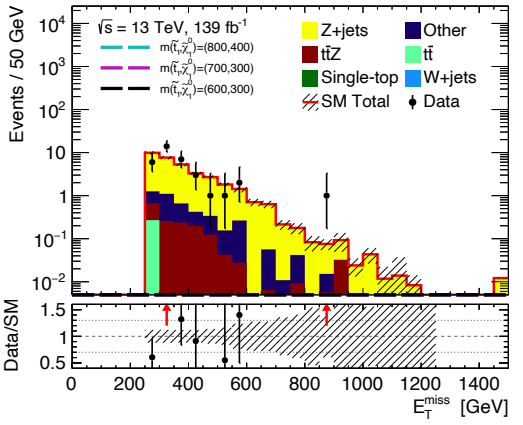
(c) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$



(d) $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$



(e) $N_{\text{b-jets}}$



(f) $E_{\mathrm{T},ll}^{\mathrm{miss}}$

Figure 9.22: Selected N-1 plots for the CRB0Z definition in (a) and (b) as well as distributions of some key variables in CRB0Z in (c)-(f), showing background and selected SRB signal benchmark points. A high purity in Z+jets events with only little contamination from other background sources is observed, while no signal contamination in the presented CR is visible. Signal point masses are given in GeV and only statistical uncertainties are shown.

Single-Top Control Region: CRAB1st

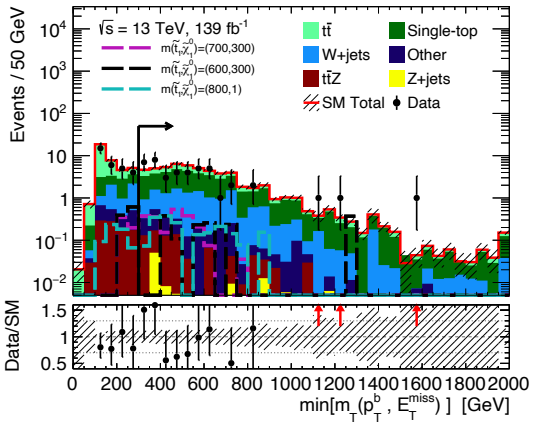
Since single-top events represent the subleading background in both SRA and SRB1, a dedicated control region, called CRAB1st, is defined in a similar way as the above discussed CRAB1Z. Table 9.14 lists the CRAB1st definition which is based on the common SRAB 1L preselection, as defined in Table 9.2. Very similar selection requirements as in its corresponding signal regions are imposed, making CRAB1st kinematically very close to both SRA and SRB1. The transverse mass requirements are slightly tightened compared to SRA and SRB1 in order to reduce the $t\bar{t}$ contamination (see Fig. 9.23(a)), whereas an upper cut on the $E_{\mathrm{T}}^{\mathrm{miss}}$ significance $\mathcal{S}$ reduces possible signal contamination (see Fig. 9.23(b)). Fig. 9.23(c) to 9.23(f) show selected distributions of some key variables in CRAB1st, where a single-top purity of 51.2% with some contamination from $t\bar{t}$ and W+jets events is observed. Table 9.15 again summarizes these background contributions and compares them to data observed in CRAB1st. The mismodeling of top background processes leads to a slight overestimation of background events compared to observed data. Furthermore, the signal contamination in CRAB1st stays at an acceptable level of below $\sim 4\%$.
Since single-top processes only represent a subleading background in SRA and SRB, and the construction of a 0L single-top VR that is close enough to the SRs to actually allow a validation of the background estimation procedure and is still dominated by single-top background processes is not feasible, no dedicated single-top VR will be defined.

Variable	**CRAB1st**
Preselection	SRAB 1L preselection
$N_{\text{top-jets}}$	≥ 1
N_{jets}	≥ 3
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	> 300
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	> 300
$m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ [GeV]	> 100
$\mathcal{S}$	$[12, 22]$
m_{T2} [GeV]	> 200

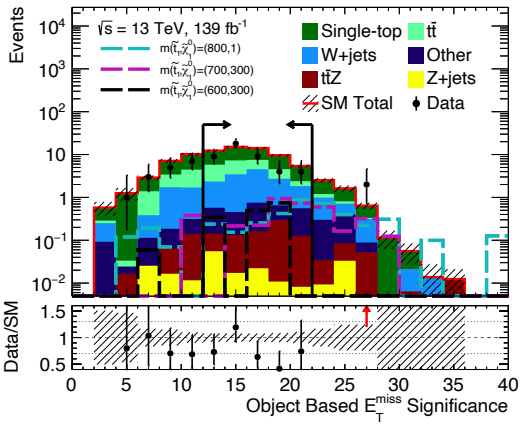
Table 9.14: Definition of CRAB1st on top of the SRAB 1L preselection defined in Table 9.2.

Process	**CRAB1st**
Single-top	29.03 ± 1.02 (51.2%)
$t\bar{t}$	12.66 ± 0.42 (22.4%)
W+jets	11.91 ± 0.96 (21.1%)
Other	2.13 ± 0.31 (3.8%)
$t\bar{t}Z$	0.74 ± 0.12 (1.3%)
Z+jets	0.11 ± 0.05 (0.2%)
SM	56.60 ± 1.50
Data	44.00 ± 6.63
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (700, 300)$ GeV	2.12 ± 0.44
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (800, 1)$ GeV	1.82 ± 0.37
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (600, 300)$ GeV	1.58 ± 0.63

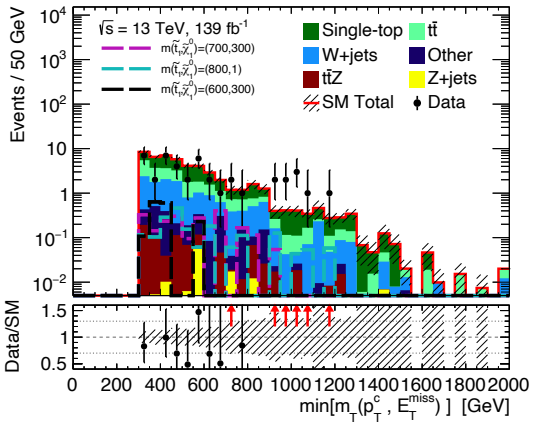
Table 9.15: Expected pre-fit SM background yields as well as data observed in CRAB1st. Respective relative SM contributions to the total CRAB1st yield are illustrated in brackets and only statistical errors are shown.



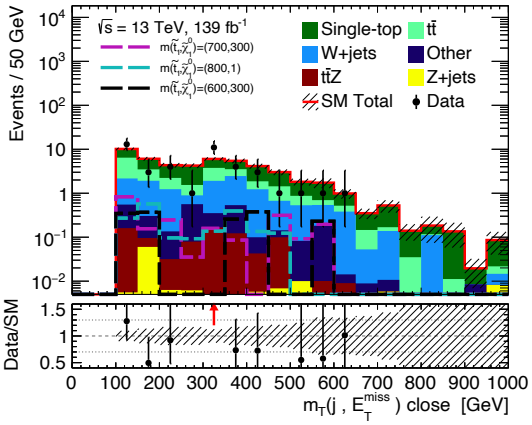
(a) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$



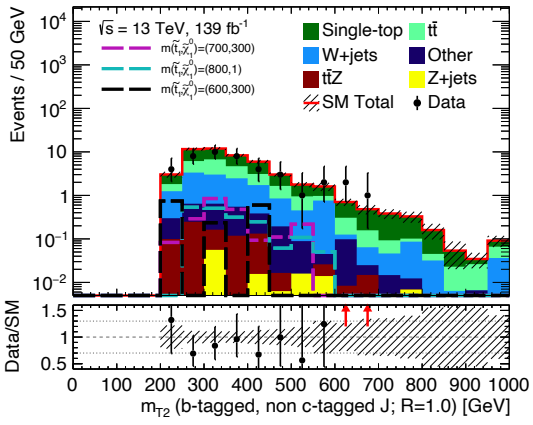
(b) $\mathcal{S}$



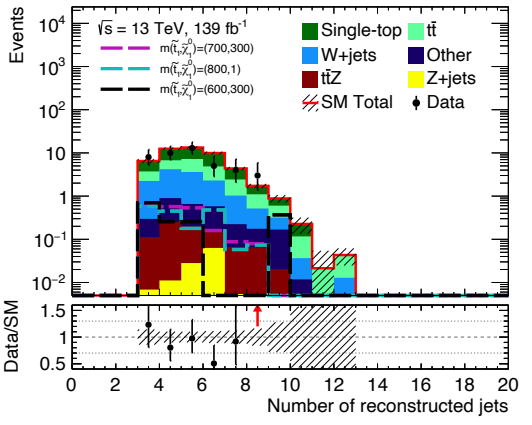
(c) $\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$



(d) $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$



(e) m_{T2}



(f) N_{jets}

Figure 9.23: Selected N-1 plots for the CRAB1st definition in (a) and (b) as well as distributions of some key variables in CRAB1st in (c)-(f), showing background and selected SRA and SRB signal benchmark points. A high purity in single-top events with some contributions from other background sources is observed, while signal contamination in the presented CR is at an acceptable level. Signal point masses are given in GeV and only statistical uncertainties are shown.

Z Validation Region: VRABZ

The Z+jets background estimation procedure in SRA and SRB is validated using a dedicated Z validation region, called VRABZ. In order to define a validation region that is kinematically very close to SRA and SRB, yet orthogonal, most selection criteria are chosen to be similar to those in the SRs, while some variables used to enhance the signal purity and to reduce Z+jets contamination in the SRA and SRB definitions are inverted.
Table 9.16 shows the definition of VRABZ on top of the common SRAB 0L preselection. While the selection requirement on the $E_{\mathrm{T}}^{\mathrm{miss}}$ significance $\mathcal{S}$ ensures orthogonality to SRA and SRB1, the chosen $\max[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ cut avoids overlaps of VRABZ with SRB0. Orthogonality with all SRC regions is again guaranteed by the leading jet flavor requirement imposed during preselection. Since a top-tagged large-radius jet requirement of $N_{\text{top-jets}} \geq 1$ would again drastically reduce the available statistics, VRABZ is defined inclusively for SRA, SRB0 and SRB1 with no applied top-tag requirement.
The amount of signal events contaminating VRABZ is shown in Appendix B.6 for all considered signal points in the mass parameter plane. The maximal possible signal contamination is found in regions of high signal cross-sections with very low neutralino and stop masses that have already been excluded by pure $\mathrm{t\bar{t}} + E_{\mathrm{T}}^{\mathrm{miss}}$ and $\mathrm{c\bar{c}} + E_{\mathrm{T}}^{\mathrm{miss}}$ searches. The signal contamination stays well below 5% for most signal points relevant in the presented analysis and does not exceed 15% in the relevant region of lower stop and neutralino masses.

Variable	**VRABZ**
Preselection	SRAB 0L preselection
N_{jets}	[3-8]
$p_{\mathrm{T}}^{b_1}$ [GeV]	> 200
$p_{\mathrm{T}}^{c_1}$ [GeV]	< 200
$N_{\text{top-jets}}$	–
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	> 200
$\min[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{b}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	> 200
$\max[m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{c}}, E_{\mathrm{T}}^{\mathrm{miss}})]$ [GeV]	< 400
$m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\text{close}}$ [GeV]	> 150
$\mathcal{S}$	[15,17]

Table 9.16: Definition of VRABZ on top of the common SRAB 0L preselection.

9.8.2 Background Estimation in Signal Region C

The following briefly discusses the control and validation region definitions to normalize the dominant background processes $\mathrm{t\bar{t}}$, W+jets and Z+jets in SRC.

$t\bar{t}$ and V+jets Control Regions

Table 9.17 shows the definition of the SRC control regions for $t\bar{t}$, similar to SRC subdivided into several m_{eff} bins[2], as well as V+jets backgrounds. All CRs are kinematically close to SRC by requiring a neural network score of $\text{NN}_{\text{score}} > 0.0$ and $\Delta R(j_1, j_2) > 1.0$, while orthogonality is ensured by the 1L and 2L preselection requirements.
In order to reduce the $t\bar{t}$ contamination in CRCW, a minimum distance selection between the leading b-jets and leading lepton $\Delta R(b_1, l_1)$ is imposed and the multi-class classification feature of the neural network is used to require a minimum V+jets neural network score. As an additional cut on the $t\bar{t}$ neural network score $\text{NN}^{t\bar{t}}_{\text{score}}$ was not able to further increase the already high purity in the SRC $t\bar{t}$ control regions, no requirements are imposed on this variable.
The expected pre-fit Standard Model background and signal yields in all SRC control regions are listed in Table 9.18.

Variable	**CRCtt750**	**CRCtt1000**	**CRCtt1250**	**CRCW**	**CRCZ**
m_{eff} [GeV]	$[750, 1000)$	$[1000, 1250)$	> 1250	–	
Preselection	SRC 1L				SRC 2L
NN_{score}	> 0.0				
$\Delta R(j_1, j_2)$	> 1.0				
$N_{\text{b-jets}}$	≥ 2			1	≥ 1
$\min[m_{\text{T}}(p_{\text{T}}^{\text{c}}, E_{\text{T}}^{\text{miss}})]$ [GeV]	> 100			> 150	> 100
$\Delta R(b_1, l_1)$	–			> 1.8	–
$\text{NN}^{\text{V+jets}}_{\text{score}}$	–			> 0.0	–
$p_{\text{T}}^{l_2}$ [GeV]	–				> 30

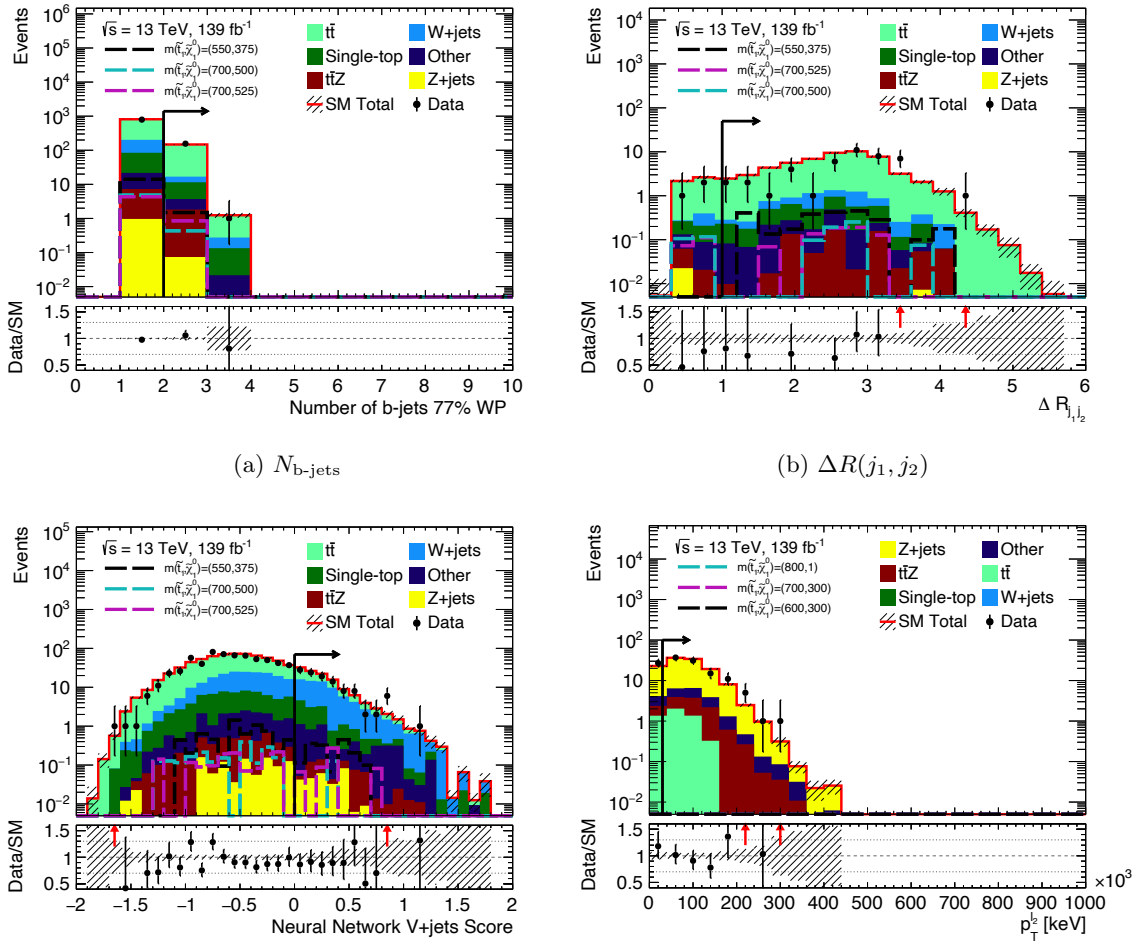
Table 9.17: Definition of SRC control regions on top of the SRC 1L and 2L preselection.

Process	**CRCtt750**	**CRCtt1000**	**CRCtt1250**	**CRCW**	**CRCZ**
Z+jets	0.07 ± 0.04	0.10 ± 0.05	0.01 ± 0.01	0.29 ± 0.08	89.62 ± 1.88
Other	2.04 ± 0.23	1.74 ± 0.25	1.23 ± 0.23	4.76 ± 0.71	10.33 ± 0.54
$t\bar{t}Z$	1.75 ± 0.20	1.48 ± 0.18	0.67 ± 0.16	0.42 ± 0.10	7.59 ± 0.23
$t\bar{t}$	131.77 ± 1.98	89.12 ± 1.51	48.77 ± 0.88	50.03 ± 1.13	3.64 ± 0.70
W+jets	5.47 ± 0.85	5.08 ± 0.69	3.09 ± 0.44	58.12 ± 2.87	0.00 ± 0.00
Single-top	7.91 ± 0.84	4.90 ± 0.54	2.64 ± 0.28	10.52 ± 0.84	0.00 ± 0.00
SM	149.01 ± 2.34	102.42 ± 1.77	56.42 ± 1.06	124.14 ± 3.27	111.18 ± 2.08
Data	157.00 ± 12.53	102.00 ± 10.10	42.00 ± 6.48	112.00 ± 10.58	107.00 ± 10.34
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (550, 375)$	1.49 ± 0.53	0.75 ± 0.29	2.53 ± 0.60	1.62 ± 0.45	0.00 ± 0.00
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (700, 525)$	0.85 ± 0.28	0.56 ± 0.25	0.71 ± 0.26	0.51 ± 0.21	0.00 ± 0.00
$m(\tilde{t}_1, \tilde{\chi}_1^0) = (700, 500)$	0.43 ± 0.18	0.18 ± 0.13	0.62 ± 0.24	0.13 ± 0.13	0.00 ± 0.00

Table 9.18: Expected pre-fit SM background yields as well as data observed in all SRC control region. Signal point masses are given in GeV and only statistical uncertainties are shown.

[2]SRC bins with $m_{\text{eff}} > 1250\,\text{GeV}$ are combined in CRCtt1250 to increase the available statistics.

High background purities of up to 87.7% and 81.8% are achieved for the $t\bar{t}$ and Z+jets backgrounds in their respective CRs, while CRCW suffers from large $t\bar{t}$ contamination and achieves a W+jets purity of 46.3%. Contamination from signal events is observed only at a negligible level. Selected N-1 plots for control regions CRCtt750, CRCtt1250, CRCW as well as CRCZ are shown in Fig. 9.24.



(a) $N_{\text{b-jets}}$ (b) $\Delta R(j_1, j_2)$

(c) $\text{NN}_{\text{score}}^{\text{V+jets}}$ (d) $p_{\text{T}}^{l_2}$ [GeV]

Figure 9.24: Selected N-1 plots showing background and selected SRC signal benchmark points for regions (a) CRCtt750, (b) CRCtt1250, (c) CRCW and (d) CRCZ. Signal point masses are given in GeV and only statistical uncertainties are shown.

Validation Regions for SRC

In order to validate the $t\bar{t}$, W+jets and Z+jets normalization factors extracted from the above described control regions, dedicated validation regions VRC are defined in Table 9.19 on top of the SRC 0L preselection.

The modeling of all relevant background processes is simultaneously validated in VRC, while again multiple m_{eff} bins are considered to properly account for the different $t\bar{t}$ control region bins. An event selection close but orthogonal to SRC is guaranteed by the imposed NN_{score} requirement and an additional upper $E_{\text{T}}^{\text{miss}}$ limit is set in VRC1750 in order to reduce possible signal contamination to levels of less than 15%.

Variable	VRC750	VRC1000	VRC1250	VRC1500	VRC1750
m_{eff} [GeV]	$[750, 1000)$	$[1000, 1250)$	$[1250, 1500)$	$[1500, 1750)$	> 1750
$E_{\text{T}}^{\text{miss}}$ [GeV]	–				< 600
Preselection	SRC 0L preselection				
NN_{score}	[0.0,0.5]				
$\Delta R(j_1, j_2)$	> 1.0				

Table 9.19: Definition of SRC validation regions on top of the SRC 0L preselection defined in Table 9.3.

9.9 Systematic Uncertainties

The estimation of background and signal yields is highly affected by systematic uncertainties which are accounted for as nuisance parameters during the maximum likelihood fit, as discussed in Section 8.2.1. Systematic uncertainties can be categorized as experimental uncertainties, for example originating from finite detector resolution effects, as well as theoretical uncertainties arising from the modeling of physics processes during Monte Carlo simulation. The following briefly discusses the most important systematic uncertainties relevant for the presented analysis.

9.9.1 Experimental Uncertainties

Systematic experimental uncertainties are associated to the methods used to reconstruct, identify and calibrate the physics objects used in the analysis. They are evaluated by computing the expected event numbers when performing up and down variations of the nominal samples based on Monte Carlo weights applied at event or object level.

Jet Energy and Mass

Two major sources of systematic experimental uncertainties result from the calibration of the jet energy scale (JES) and jet energy resolution (JER), as already discussed in Section 9.4.3. The JES uncertainty is derived in bins of η and p_{T} using *in-situ* techniques, which exploit the transverse momentum balance between jets and well-calibrated reference objects such as a Z-bosons, while also uncertainties related to jet flavor composition and pile-up effects are considered [157]. While a full list of 125 JES nuisance parameters is available and could be included in the fit, only a reduced set of 12 orthogonal components is used instead, as it drastically reduces computational needs while the impact on physics results is negligible.
Similar to the JES uncertainty, systematic uncertainties on the JER are evaluated from differences between data and Monte Carlo simulations and in-situ measurements of transverse momentum balances between jets and reference objects. A reduced nuisance parameter set including 11 JER uncertainty parameters is used [157].
Large-radius JES and JER uncertainties are derived in the same fashion as explained above for their $R = 0.4$ counterparts and are included in the fit as a reduced set of 20 nuisance parameters. Additionally, uncertainties arising during the calibration of the large-radius

jet mass to correct the calorimeter response to the true jet mass are taken into account. The evaluation of the large-radius jet mass scale (JMS) and jet mass resolution (JMR) uncertainties is also based on in-situ techniques using measurements of well-defined reference objects, as explained in full detail in Ref. [169]. Nine additional nuisance parameters are incorporated in the fit to account for the large-radius JMS and JMR uncertainties.
As systematic uncertainties for small- and large-radius jets might exhibit correlations, their corresponding nuisance parameters are treated as being correlated in the statistical fit framework.

Flavor Tagging

This analysis heavily relies on the identification of b-jets, c-jets as well as large-radius top-jets. All of the employed heavy-flavor tagging algorithms entail systematic uncertainties which need to be considered. The b-tagging uncertainties are given as a reduced nuisance parameter set consisting of scale factor uncertainties (see Section 9.4.6) for b-tagging efficiency, c-jet mistag rate, light-jet mistag rate and high-p_{T} extrapolation [166].
While the top-tagging systematic uncertainties are treated analogously by including dedicated nuisance parameters to the likelihood fit, accounting for scale factor uncertainties for the top-tagging efficiency as well as for the respective mistag rates [162, 163], scale factor uncertainties for the deployed c-tagging algorithm are not yet available at the time of writing. Thus, a conservative 30% systematic uncertainty on the c-tagging scale factor is used instead.

Missing Transverse Momentum

The $E_{\mathrm{T}}^{\mathrm{miss}}$ systematic uncertainty is evaluated based on the scale and resolution uncertainties of all individual input objects that are used to derive the $E_{\mathrm{T}}^{\mathrm{miss}}$ quantity as defined in Eq. (9.2) and included in the fit as dedicated nuisance parameters. Additional uncertainties related to the scale and resolution of soft term in Eq. (9.2) have been derived using in-situ techniques with $\mathrm{Z} \to \mu\mu$ reference events (see Ref. [170]) and propagated to the $E_{\mathrm{T}}^{\mathrm{miss}}$ systematic uncertainty calculation.

Leptons

As leptons are vetoed in the signal regions of the analysis, lepton-related uncertainties will only contribute to the total uncertainty through their impact on the background estimation using 1L and 2L control regions. Dedicated nuisance parameters for the systematic uncertainties on the electron and muon energy resolution, reconstruction and identification as well as trigger efficiencies are considered and included in the likelihood fit.

Luminosity and Pile-up Reweighting

The total integrated luminosity for pp collisions at $\sqrt{s} = 13\,\mathrm{TeV}$ recorded by the ATLAS detector during Run 2 corresponds to $139\,\mathrm{fb}^{-1}$, measured with an uncertainty of $\pm 1.7\%$ [171]. A dedicated nuisance parameter is added to the fit to allow for up and down

variations of the total integrated luminosity to account for these systematic uncertainties. Lastly, uncertainties related to the reweighting of Monte Carlo events to match the pile-up distributions found in data need to be taken into account for which a nuisance parameter is added to the fit, allowing for variations of the applied pile-up event weight [172].

9.9.2 Theoretical Uncertainties

As already discussed in Section 3.4, simulated Monte Carlo events are typically based on a variety of different parameters and scales introduced during the event generation stage, highly affecting simulation steps like the matrix element calculation, parton shower and hadronization model as well as the simulation of initial and final state radiation. In order to evaluate the impact of the associated systematic uncertainties arising from the choice of the employed generators, scales and parameter sets, the configurations used to produce the nominal background and signal samples are usually systematically varied and the full MC event simulation production is repeated to produce new samples which can be compared to the nominal ones. The following briefly discusses these theoretical uncertainties originating from various generator modeling effects for important background and signal processes, while uncertainties on background processes not covered in the following are negligible and therefore not taken into account during the fitting procedure.

V+Jets Modeling Uncertainties

Major uncertainties affecting the V+jets modeling are the renormalization scale μ_R and factorization scale μ_F uncertainties as well as the matrix element CKKW matching scale and resummation scale μ_Q uncertainties. While both μ_R and μ_F are simultaneously varied by a factor $\frac{1}{2}$, 1 (i.e. the nominal scale) or 2, resulting in a total of six pairwise combinations to evaluate the μ_R and μ_F uncertainties, the CKKW matching scale is modified between 15 GeV and 30 GeV with its nominal value at 20 GeV and the resummation scale μ_Q is varied by a factor of $1/\sqrt{2}$ and $\sqrt{2}$.

As Z+jets events present the most important background in SRA, SRB0 and SRB1, additional studies were conducted to ensure an adequate modeling of these processes, in particular the production of Z bosons in association with heavy-flavor jets, and their systematic uncertainties in the phase space regions of interest. Details on Z+jets flavor composition studies in the SRs, CRs and VRs can be found in Appendix B.7.1.

Ref. [173] recently studied the predictions of Z boson production in association with b-jets in proton-proton collisions at a center-of-mass energy of 13 TeV for various MC generators, including the Sherpa 5FNS generator used in this analysis. It was found that Sherpa 5FNS delivers the best accuracy in describing the observed data and generally describes various measured differential distributions within the modeling uncertainties. Good agreement with data within the relatively large uncertainties of up to 30% for Z+$\geq$ 1b-jet production and up to 50% for Z+$\geq$ 2b-jet production was also seen for large values of b-jet transverse momenta, a phase space region of particular interest for this analysis. Comparably large uncertainties, especially caused by the renormalization and factorization scale uncertainties

described above, have been observed in this analysis' CRs, VRs and SRs for various b- and c-jet kinematic distributions. However, due to a very similar behavior of these systematic uncertainties in the SRs and their respective CRs, a large cancellation by virtue of the transfer factor approach is expected during the fit.
In addition to events involving Z production in association with up to two heavy-flavor quarks (i.e. one truth-level $c\bar{c}$ or truth-level $b\bar{b}$ pair), also processes of Z production in association with at least one truth-level $c\bar{c}$ *and* at least one truth-level $b\bar{b}$ pair, referred to as Z+$c\bar{c}b\bar{b}$ in the following, contribute to the overall Z+jets background in SRA, SRB0 and SRB1[3]. Given that these Z+$c\bar{c}b\bar{b}$ events, involving the production of at least four heavy-flavor jets, present a rather unusual (and thus potentially not yet very well understood) background process, more detailed studies to assess the uncertainty on Z+$c\bar{c}b\bar{b}$ events have been conducted, as briefly discussed in Appendix B.7.2.
In order to better account for potential mismodeling of this background component, it was decided to introduce a conservative 107% systematic uncertainty on events containing Z+$c\bar{c}b\bar{b}$ processes in all SRs.
In addition to the aforementioned V+jets modeling uncertainties, PDF uncertainties on the gluon, light-quark and heavy-flavor quark PDFs are taken into account for V+jets processes by comparing the nominal NNPDF3.0 [133] PDF set with the CT14 [174] and MMHT14 [175] PDF sets, following the current ATLAS and PDF4LHC recommendations for LHC Run 2 [176].

$t\bar{t}$ Modeling Uncertainties

Systematic uncertainties affecting $t\bar{t}$ processes also involve the above described scale uncertainties of the hard-scatter modeling, uncertainties due to the employed hadronization model as well as initial and final state radiation uncertainties. Hard-scatter and parton shower uncertainties are evaluated using alternative $t\bar{t}$ samples generated with Powheg-Box interfaced with Herwig 7 [177] and MadGraph5_aMC@NLO interfaced with Pythia 8 in the A14 tune. Following the default recommendation, initial and final state radiation uncertainties are evaluated by reweighting the nominal $t\bar{t}$ samples as described in Ref. [178]. PDF uncertainties were found to be very small and are therefore neglected.

Single-Top Modeling Uncertainties

Single-top modeling uncertainties are very similar to the above described $t\bar{t}$-related uncertainties. Hard-scatter and shower uncertainties are again evaluated using alternative simulation samples, produced using the same alternative generators as above, while initial and final state radiation uncertainties are also determined based on a reweighting of the nominal events. Single-top background modeling introduces an additional source of systematic uncertainty originating from the possible interference of single-top W+t samples with $t\bar{t}$: the generation of W+t events at NLO results in diagrams that overlap with $t\bar{t}$

[3] Z+$c\bar{c}b\bar{b}$ events make up about 23% of the Z+jets background in SRA, 16% of the Z+jets background in SRB0 as well as 26% of the Z+jets background in SRB1.

LO production diagrams which need to be accounted for in order to avoid double counting [178]. The nominal W+t samples use the *diagram removal* (DR) scheme while the alternative samples are based on the *diagram subtraction* (DS) scheme [178]. The uncertainty arising due to the difference between the DR and DS schemes is expected to be the dominant source of uncertainty for single-top processes, as detailed in Appendix B.8.

Signal Uncertainties

The signal modeling is affected by theoretical uncertainties very similar to those relevant in the modeling of Standard Model background processes. While factorization, renormalization, resummation as well as CKKW matching scale uncertainties are evaluated using internal weighting factor provided with the nominal signal samples, alternative signal samples are used to determine initial and final state radiation uncertainties in signal processes.

9.10 Results

This section presents the results obtained from performing different maximum likelihood fits as well as hypothesis tests in the tc + $E_{\mathrm{T}}^{\mathrm{miss}}$ analysis. After a background-only fit is performed in the control regions to estimate the background contributions, the background estimation strategy is subsequently validated in the validation regions and the signal regions are unblinded to compare the observed data with the Standard Model expectations.

9.10.1 Background-only Fit Results

Results in Control Regions

A background-only fit is performed in all control regions simultaneously, allowing to extract normalization factors to fit the dominant background processes to data. The resulting normalization factors and their uncertainties are presented in Table 9.20. Most background processes roughly stay at their nominal MC expectation values with their respective normalization factors close to unity, while in particular single-top processes in SRA and SRB1 as well as $\mathrm{t\bar{t}}$ processes with $m_{\mathrm{eff}} > 1250\,\mathrm{GeV}$ in SRC are scaled down considerably, a behavior that is also seen in several other ATLAS analyses (see e.g. Refs. [179, 180]). The high relative uncertainties on $\mu_{\mathrm{st,AB1}}$ and $\mu_{\mathrm{W,C}}$ can mostly be attributed to the comparably low purity of single-top and W+jets events in CRAB1st and CRCW, respectively.
Table 9.21 summarizes the expected and observed event yields in all CRs after the background-only fit. By construction, a good agreement between the fitted MC background yields and data is observed. Example distributions of representative variables in CRAB1Z and CRB0Z after the background-only fit are shown in Fig. 9.25. Although single normalization factors are used, which can only account for differences in the overall normalization and are not able to alter the shape of the respective distributions, a good agreement between the fitted MC background estimation and observed data can be seen.
A similarly good agreement of the post-fit MC background estimation with data can be observed in Fig. 9.26, presenting CRCW, CRCZ as well as CRCtt750 and CRCtt1250 example distributions of important kinematic quantities.

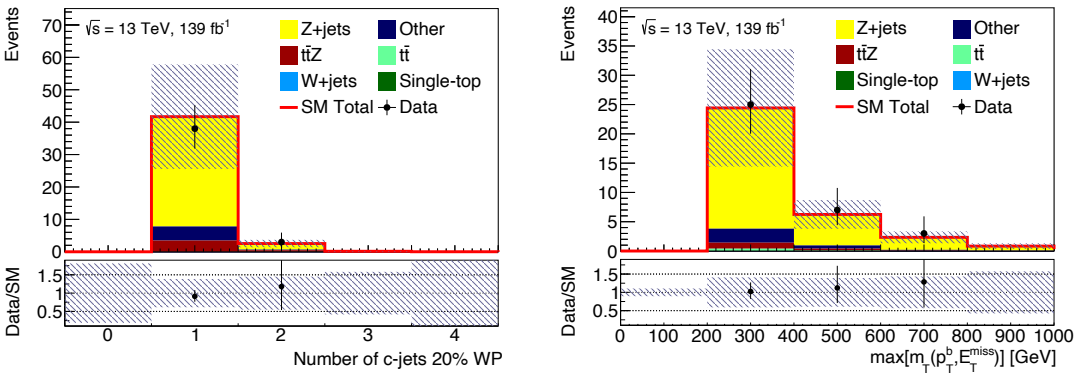
Normalization factor	Value
$\mu_{\mathrm{Z,AB}}$	0.91 ± 0.16
$\mu_{\mathrm{st,AB1}}$	0.57 ± 0.33
$\mu_{\mathrm{W,C}}$	0.95 ± 0.41
$\mu_{\mathrm{Z,C}}$	0.96 ± 0.12
$\mu_{\mathrm{t\bar{t},750}}$	1.07 ± 0.12
$\mu_{\mathrm{t\bar{t},1000}}$	0.99 ± 0.13
$\mu_{\mathrm{t\bar{t},1250}}$	0.71 ± 0.18

Table 9.20: Normalization factors extracted from the background-only fit for Z+jets ($\mu_{\mathrm{Z,AB}}$ and $\mu_{\mathrm{Z,C}}$), single-top ($\mu_{\mathrm{st,AB1}}$), W+jets ($\mu_{\mathrm{W,C}}$) as well as $\mathrm{t\bar{t}}$ ($\mu_{\mathrm{t\bar{t},750}}$, $\mu_{\mathrm{t\bar{t},1000}}$ and $\mu_{\mathrm{t\bar{t},1250}}$) processes.

Region	CRAB1Z	CRB0Z	CRAB1st	CRCtt750
Observed events	41	35	44	157
Fitted SM events	42.69 ± 8.49	33.41 ± 6.85	43.98 ± 9.71	157.00 ± 17.12
Fitted Z+jets	34.15 ± 6.54	28.39 ± 5.66	0.10 ± 0.08	0.07 ± 0.04
Fitted W+jets	$0.00^{+0.02}_{-0.00}$	$0.00^{+0.01}_{-0.00}$	11.63 ± 6.04	5.11 ± 2.37
Fitted $t\bar{t}$	$0.00^{+0.06}_{-0.00}$	$0.39^{+0.83}_{-0.39}$	12.98 ± 6.46	139.83 ± 13.14
Fitted single-top	$0.00^{+0.03}_{-0.00}$	$0.07^{+0.28}_{-0.07}$	16.43 ± 9.54	8.22 ± 3.12
Fitted $t\bar{t}Z$	3.95 ± 1.92	1.44 ± 1.21	0.73 ± 0.41	1.78 ± 0.22
Fitted other	4.59 ± 3.32	3.12 ± 2.01	2.11 ± 0.39	2.01 ± 0.53
Pre-fit SM	47.78 ± 1.33	35.58 ± 1.25	56.60 ± 1.50	149.01 ± 2.34

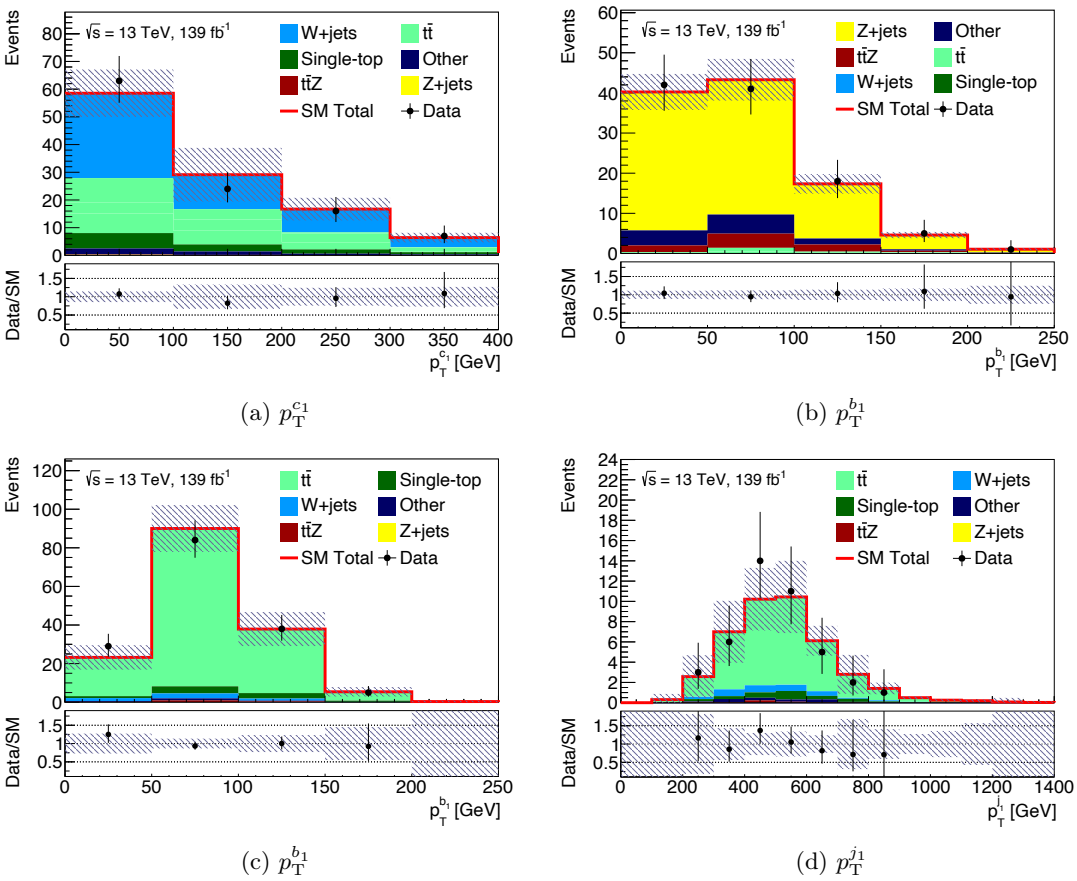
Region	CRCtt1000	CRCtt1250	CRCW	CRCZ
Observed events	102	42	112	107
Fitted SM events	102.01 ± 12.09	42.00 ± 7.46	112.09 ± 19.56	107.00 ± 11.84
Fitted Z+jets	0.10 ± 0.08	$0.01^{+0.01}_{-0.01}$	0.28 ± 0.11	86.56 ± 10.61
Fitted W+jets	4.82 ± 2.03	2.99 ± 1.85	55.43 ± 23.37	$0.00^{+0.01}_{-0.00}$
Fitted $t\bar{t}$	89.04 ± 10.68	34.41 ± 6.97	40.91 ± 10.61	2.45 ± 0.90
Fitted single-top	4.82 ± 2.84	2.67 ± 1.54	10.37 ± 9.04	0.00 ± 0.13
Fitted $t\bar{t}Z$	1.53 ± 0.20	0.68 ± 0.19	0.42 ± 0.10	7.56 ± 1.12
Fitted other	1.70 ± 0.49	1.24 ± 0.23	4.68 ± 1.38	10.43 ± 1.28
Pre-fit SM	102.42 ± 1.77	56.42 ± 1.06	124.14 ± 3.27	111.18 ± 2.08

Table 9.21: Background-only fit results in all CRs for an integrated luminosity of 139 fb^{-1}. The presented errors include both statistical and systematic uncertainties. For comparison, the respective total pre-fit MC expectations with their statistical uncertainties are shown.



(a) $N_{\text{c-jets}}$ (b) $\max[m_{\text{T}}(p_{\text{T}}^{\text{b}}, E_{\text{T}}^{\text{miss}})]$

Figure 9.25: Representative distributions of selected variables shown in (a) CRAB1Z and (b) CRB0Z after the background-only fit. The presented errors include both statistical as well as systematic uncertainties. Good agreement between post-fit MC expectation and data is observed.



(a) $p_{\mathrm{T}}^{c_1}$ (b) $p_{\mathrm{T}}^{b_1}$

(c) $p_{\mathrm{T}}^{b_1}$ (d) $p_{\mathrm{T}}^{j_1}$

Figure 9.26: Representative distributions of selected variables shown in (a) CRCW, (b) CRCZ, (c) CRCtt750 and (d) CRCtt1250 after the background-only fit. The presented errors include both statistical as well as systematic uncertainties. Good agreement between post-fit MC expectation and data is observed.

Results in Validation Regions

In order to gain trust in the background estimation strategy, the background-only fit results in the CRs are first validated by extrapolating them to their respective VRs. Fitted SM background estimates and observed data yields for the validation regions VRABZ as well as all VRCs are presented in Table 9.22. Good agreement within statistical and systematic uncertainties between data and fitted SM background yields can be seen in all regions, while a light overfluctuation in data with a significance of $1.25\,\sigma$ is observed in VRC1500. All significance values discussed here as well as in all following sections are derived as recommended in Ref. [181].

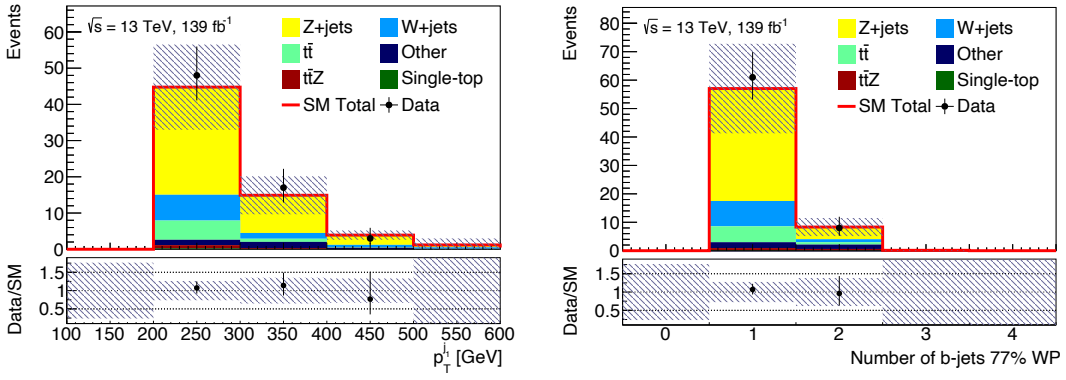
Fig. 9.27 shows example distributions of representative variables in VRABZ after the background-only fit, where a good agreement between the post-fit MC background expectation and data is observed. Furthermore, a similarly good agreement can also be observed in VRC750 and VRC1000, as illustrated in Fig. 9.28.

Post-fit MC predictions and observed data yields in all CRs and VRs of this analysis, together with their respective significances, are summarized in Fig. 9.29. The overall agreement seen in the VRs is considered to be good and allows for a reliable extrapolation of the background estimates from the CRs to the SRs in the following.

Region	VRABZ	VRC750	VRC1000
Observed events	70	1124	638
Fitted SM events	65.01 ± 13.38	1195.71 ± 152.32	679.81 ± 74.57
Fitted Z+jets	43.47 ± 15.39	160.80 ± 26.88	96.13 ± 14.16
Fitted W+jets	9.97 ± 6.28	168.22 ± 94.91	96.66 ± 50.65
Fitted $t\bar{t}$	6.34 ± 4.28	759.45 ± 122.55	422.02 ± 108.06
Fitted single-top	$0.37^{+0.62}_{-0.37}$	58.75 ± 25.70	31.08 ± 14.78
Fitted $t\bar{t}Z$	1.24 ± 0.42	16.79 ± 1.17	11.85 ± 0.96
Fitted other	3.62 ± 1.48	31.70 ± 3.60	22.07 ± 4.79
Pre-fit SM	69.83 ± 2.36	1150.21 ± 9.91	686.27 ± 4.20

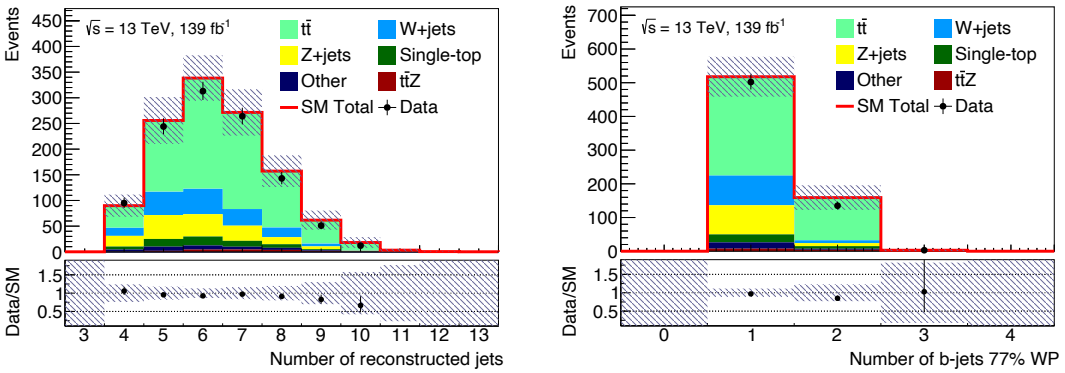
Region	VRC1250	VRC1500	VRC1750
Observed events	248	101	23
Fitted SM events	224.79 ± 31.89	79.57 ± 13.27	27.88 ± 5.73
Fitted Z+jets	39.85 ± 5.78	14.53 ± 2.55	5.54 ± 1.03
Fitted W+jets	39.26 ± 20.25	14.99 ± 6.80	4.24 ± 2.35
Fitted $t\bar{t}$	118.49 ± 37.18	38.50 ± 15.88	14.46 ± 6.72
Fitted single-top	13.25 ± 9.66	$5.58^{+5.75}_{-5.58}$	2.03 ± 1.75
Fitted $t\bar{t}Z$	4.70 ± 0.44	1.62 ± 0.49	0.53 ± 0.17
Fitted other	9.25 ± 1.16	4.35 ± 0.92	$1.07^{+1.12}_{-1.07}$
Pre-fit SM	276.98 ± 3.63	96.63 ± 2.83	34.27 ± 1.05

Table 9.22: Background-only fit results in all VRs for an integrated luminosity of 139 fb^{-1}. The presented errors include both statistical and systematic uncertainties. For comparison, the respective total pre-fit MC expectations with their statistical uncertainties are shown.



(a) $p_{\mathrm{T}}^{j_1}$ (b) $N_{\text{b-jets}}$

Figure 9.27: Representative distributions of selected variables in VRABZ after the background-only fit. The presented errors include both statistical as well as systematic uncertainties. Good agreement between post-fit MC expectation and data is observed.



(a) N_{jets} (b) $N_{\text{b-jets}}$

Figure 9.28: Representative distributions of selected variables shown in (a) VRC750 and (b) VRC1000 after the background-only fit. The presented errors include both statistical as well as systematic uncertainties. Good agreement between post-fit MC expectation and data is observed.

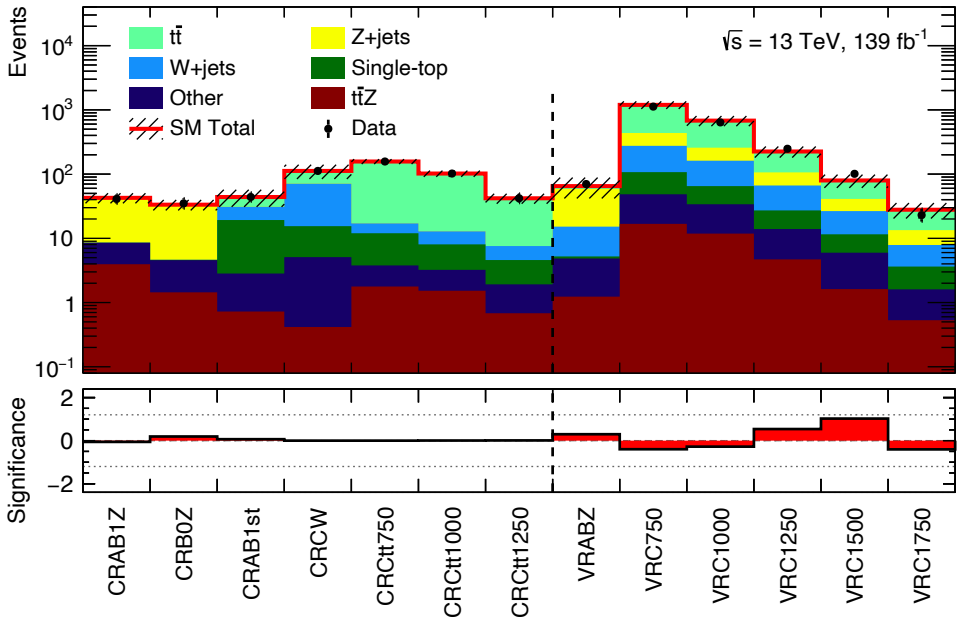


Figure 9.29: Summary of all MC predicted backgrounds and observed data events after the background-only fit (top panel), together with the resulting significances (bottom panel) in all considered CRs and VRs of this analysis.

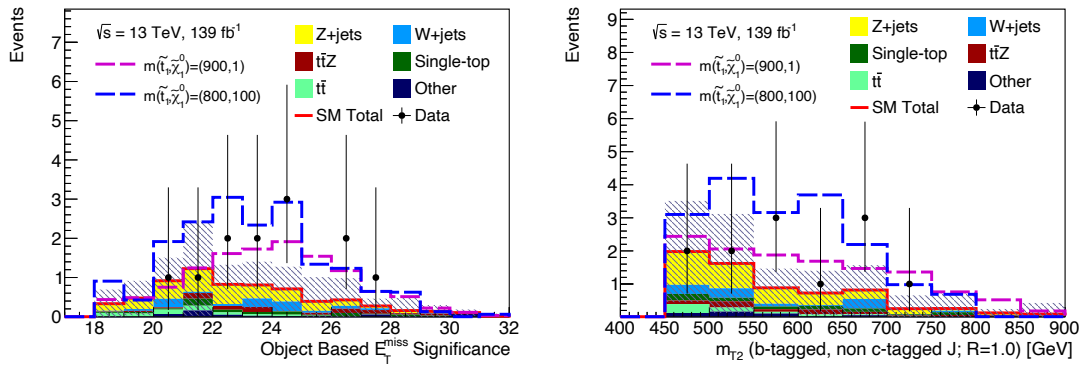
Results in Signal Regions

In the following, the background-only fit results in the CRs are extrapolated to the SRs and the SRs are unblinded to allow for a comparison of the SM background estimates with observed data.

Table 9.23 compares the SM background estimates with observed data in the two SRA bins. Z+jets is the most dominant background in both bins, followed by W+jets as well as single-top processes which contribute less compared to the pre-fit estimates due to the relatively low single-top normalization factor. Total systematic and statistical uncertainties amount to more than 50% of the total SM yield, while for some processes, especially $t\bar{t}$ and single-top production, large uncertainties of well above 100% can be seen, which mainly arise from large fluctuations observed in variations of relevant systematic quantities in the considered phase space region. The impact of the most dominant systematic uncertainties of all regions will be discussed later in more detail.
While the data observed in SRA-450 is well within 1σ of the SM prediction, the SRA-575 bin shows a slight overfluctuation of about 1.3σ. Two example post-fit distributions of important kinematic quantities in SRA are illustrated in Fig. 9.30, where again the slight overfluctuation for high m_{T2} values can be observed.

Region	SRA-450	SRA-575
Observed events	6	6
Fitted SM events	4.07 ± 2.24	2.88 ± 1.76
Fitted Z+jets	2.02 ± 1.22	1.25 ± 0.82
Fitted W+jets	0.62 ± 0.35	0.42 ± 0.25
Fitted $t\bar{t}$	$0.42^{+0.66}_{-0.42}$	$0.13^{+0.24}_{-0.13}$
Fitted single-top	$0.38^{+0.57}_{-0.38}$	$0.44^{+0.68}_{-0.44}$
Fitted $t\bar{t}Z$	0.32 ± 0.22	0.45 ± 0.32
Fitted other	0.30 ± 0.12	0.19 ± 0.12
Pre-fit SM	4.54 ± 0.38	3.34 ± 0.26

Table 9.23: Observed and expected SM yields in the SRA bins after the background-only fit for an integrated luminosity of 139 fb^{-1}. The number appended to the region name indicates the lower m_{T2} boundary of the SRA bin and presented post-fit errors include both statistical and systematic uncertainties. For comparison, the respective total pre-fit MC expectations with their statistical uncertainties are shown.



(a) $\mathcal{S}$ (b) m_{T2}

Figure 9.30: Representative distributions in the inclusive SRA region after the background-only fit, additionally showing two example signal scenarios (mass points given in GeV) for comparison. The shaded areas include all systematic and statistical uncertainties.

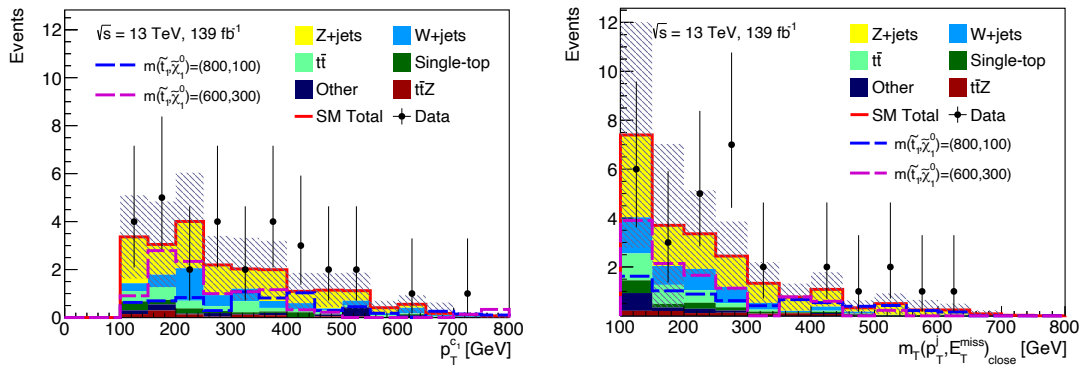
Expected SM yields and observed data in the unblinded SRB0 and SRB1 bins after the background-only fit are shown in Tables 9.24 and 9.25, respectively. The most dominant background process in all SRB0 and SRB1 bins is Z+jets production, followed by W+jets and top production processes, except for the low $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ SRB1 bin, in which Z+jets and W+jets processes contribute to roughly equal amounts. In the intermediate $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ SRB1 bins, SRB1-150 and SRB1-300, $\mathrm{t\bar{t}Z}$ becomes sub-dominant.
The combined total systematic and statistical uncertainties on the post-fit SM expectations amount to more than 50% of the total post-fit yield for most SRB0 and SRB1 bins and go as high as about 67% and 70% for the high $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ bins in SRB0 and SRB1, respectively. A detailed overview of relevant systematic uncertainties in SRB0 and SRB1 will be given below.
While the fitted SM background estimation in first two SRB0 bins shows good agreement with data within 1σ, a small excess in data can be observed in SRB0-400, quantified to about 1.6σ. Also two of the four SRB1 bins, SRB1-100 and SRB1-300, show slight data overfluctuation of about 1σ and 1.4σ, respectively, whereas good data-MC agreement can be seen in SRB1-150 and a slight underfluctuation is observed in SRB1-500.
Multiple example post-fit distributions of important kinematic quantities in SRB0 and SRB1 are presented in Fig. 9.31, again indicating the observed slight data overfluctuations in some $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ regions.

Region	SRB0-100	SRB0-150	SRB0-400
Observed events	6	17	7
Fitted SM events	7.33 ± 4.51	11.52 ± 5.89	2.55 ± 1.71
Fitted Z+jets	3.31 ± 2.06	5.65 ± 2.95	1.80 ± 1.31
Fitted W+jets	1.48 ± 0.86	2.64 ± 1.45	0.34 ± 0.21
Fitted $\mathrm{t\bar{t}}$	$1.08^{+1.22}_{-1.08}$	$1.53^{+1.84}_{-1.53}$	$0.06^{+0.11}_{-0.06}$
Fitted single-top	$0.53^{+0.66}_{-0.53}$	$0.65^{+0.71}_{-0.65}$	$0.14^{+0.18}_{-0.14}$
Fitted $\mathrm{t\bar{t}Z}$	0.20 ± 0.18	0.62 ± 0.41	$0.08^{+0.11}_{-0.08}$
Fitted other	0.73 ± 0.65	0.43 ± 0.38	0.12 ± 0.09
Pre-fit SM	8.04 ± 0.84	11.84 ± 1.08	2.64 ± 0.49

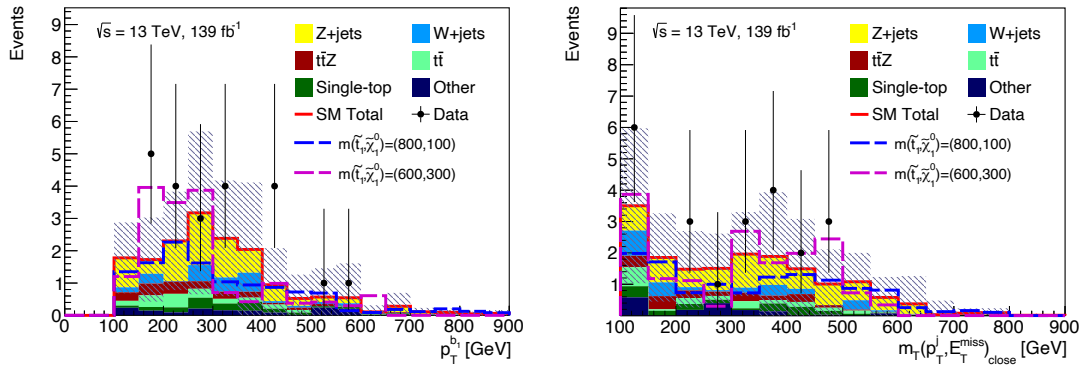
Table 9.24: Observed and expected SM yields in the SRB0 bins after the background-only fit for an integrated luminosity of $139\,\mathrm{fb}^{-1}$. The number appended to the region name indicates the lower $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ boundary of the SRB0 bin and presented post-fit errors include both statistical and systematic uncertainties. For comparison, the respective total pre-fit MC expectations with their statistical uncertainties are shown.

Region	SRB1-100	SRB1-150	SRB1-300	SRB1-500
Observed events	6	4	12	0
Fitted SM events	3.49 ± 2.13	4.78 ± 2.61	6.25 ± 3.41	2.22 ± 1.55
Fitted Z+jets	0.78 ± 0.54	2.04 ± 1.24	3.30 ± 2.03	1.42 ± 1.07
Fitted W+jets	0.80 ± 0.54	0.59 ± 0.38	0.53 ± 0.35	0.41 ± 0.25
Fitted $t\bar{t}$	$0.60^{+0.69}_{-0.60}$	$0.34^{+0.70}_{-0.34}$	$0.59^{+0.77}_{-0.59}$	$0.14^{+0.31}_{-0.14}$
Fitted single-top	$0.36^{+0.40}_{-0.36}$	$0.48^{+0.54}_{-0.48}$	$0.75^{+0.77}_{-0.75}$	$0.11^{+0.37}_{-0.11}$
Fitted $t\bar{t}Z$	0.36 ± 0.26	0.80 ± 0.45	0.74 ± 0.31	$0.04^{+0.06}_{-0.04}$
Fitted other	0.58 ± 0.55	0.53 ± 0.28	0.35 ± 0.24	0.09 ± 0.06
Pre-fit SM	3.99 ± 0.45	5.27 ± 0.59	7.38 ± 0.82	2.46 ± 0.41

Table 9.25: Observed and expected SM yields in the SRB1 bins after the background-only fit for an integrated luminosity of 139 fb^{-1}. The number appended to the region name indicates the lower $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$ boundary of the SRB1 bin and presented post-fit errors include both statistical and systematic uncertainties. For comparison, the respective total pre-fit MC expectations with their statistical uncertainties are shown.



(a) $p_{\mathrm{T}}^{c_1}$ (b) $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$



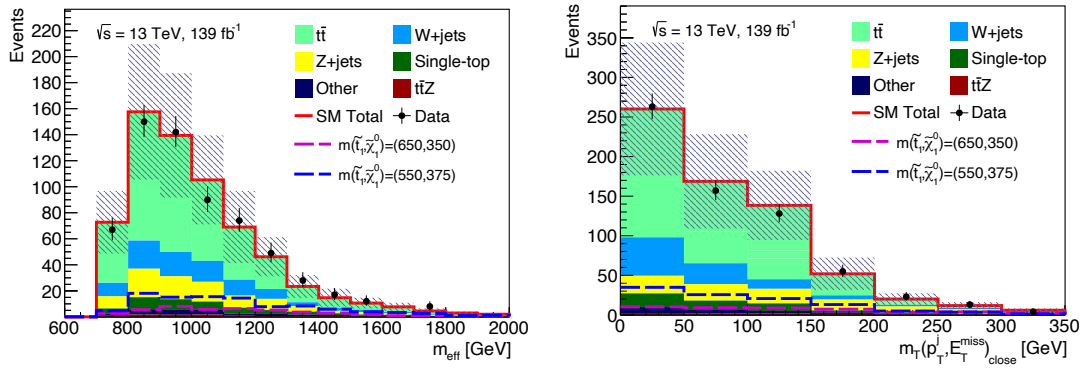
(c) $p_{\mathrm{T}}^{b_1}$ (d) $m_{\mathrm{T}}(p_{\mathrm{T}}^{\mathrm{j}}, E_{\mathrm{T}}^{\mathrm{miss}})_{\mathrm{close}}$

Figure 9.31: Representative distributions in the inclusive SRB0 region in (a) and (b) as well as in the inclusive SRB1 region in (c) and (d) after the background-only fit, additionally showing two example signal scenarios (mass points given in GeV) for comparison. The shaded areas include all systematic and statistical uncertainties.

Lastly, observed data and expected SM yields in the SRC m_{eff} bins after the background-only fit are presented in Table 9.26. Note that, for conciseness, only the inclusive SRC m_{eff} bins, rather than the two-dimensional SRC binning in m_{eff} and $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$, is shown. In all SRC bins, $\text{t}\bar{\text{t}}$ production represents the most dominant background process, followed by both Z+jets and W+jets processes which contribute to roughly equal amounts. Total systematic and statistical uncertainties stay at around $30-35\%$ for small m_{eff} values and increase to up to 60% of the post-fit SM prediction in the SRC-2000 bin. Again, a more detailed discussion of the most dominant systematic uncertainties will be given below.
All observed data yields, both in the shown SRC m_{eff} bins as well as in the not here depicted $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$ sub-bins, are found to be in good agreement with the SM prediction. Post-fit distributions of the m_{eff} and $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$ variables in SRC are illustrated in Fig. 9.32, again showing the good data-MC agreement.

Region	SRC-750	SRC-1000	SRC-1250
Observed events	359	195	63
Fitted SM events	369.38 ± 120.86	202.81 ± 67.57	55.60 ± 19.85
Fitted $\text{t}\bar{\text{t}}$	235.05 ± 82.34	118.50 ± 41.39	29.09 ± 14.53
Fitted Z+jets	48.83 ± 16.28	30.46 ± 10.53	8.79 ± 3.07
Fitted W+jets	50.77 ± 33.81	32.38 ± 18.48	8.65 ± 6.22
Fitted single-top	20.88 ± 13.22	12.44 ± 8.10	4.84 ± 3.01
Fitted $\text{t}\bar{\text{t}}\text{Z}$	4.26 ± 1.33	2.98 ± 1.00	1.24 ± 0.42
Fitted other	9.59 ± 4.12	6.05 ± 1.97	3.01 ± 1.15
Pre-fit SM	356.48 ± 4.97	207.36 ± 3.13	67.91 ± 1.05
Region	SRC-1500	SRC-1750	SRC-2000
Observed events	15	9	5
Fitted SM events	20.83 ± 8.83	6.99 ± 3.36	3.40 ± 2.04
Fitted $\text{t}\bar{\text{t}}$	10.16 ± 5.97	3.78 ± 1.94	1.96 ± 1.20
Fitted Z+jets	3.65 ± 1.36	0.98 ± 0.37	0.46 ± 0.17
Fitted W+jets	3.24 ± 1.84	1.12 ± 0.72	0.49 ± 0.42
Fitted single-top	2.05 ± 1.89	$0.53^{+0.63}_{-0.53}$	$0.28^{+0.42}_{-0.28}$
Fitted $\text{t}\bar{\text{t}}\text{Z}$	0.38 ± 0.21	0.10 ± 0.08	$0.02^{+0.05}_{-0.02}$
Fitted other	1.35 ± 0.44	0.47 ± 0.15	0.18 ± 0.15
Pre-fit SM	25.43 ± 0.91	8.78 ± 0.54	4.31 ± 0.26

Table 9.26: Observed and expected SM yields in the inclusive SRC m_{eff} bins after the background-only fit for an integrated luminosity of $139\,\text{fb}^{-1}$. Presented post-fit errors include both statistical and systematic uncertainties and the respective total pre-fit MC expectations with their statistical uncertainties are shown for comparison.



(a) m_{eff} (b) $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$

Figure 9.32: Representative distributions in the inclusive SRC region after the background-only fit, additionally showing two example signal scenarios (mass points given in GeV) for comparison. The shaded areas include all systematic and statistical uncertainties.

A breakdown of the most dominant systematic uncertainties after the background-only fit in the regions SRA, SRB0 and SRB1 is presented in Tables 9.27, 9.28 and 9.29, respectively, while the most dominant systematic uncertainties in SRC are listed in Appendix B.9 for the sake of conciseness.

In all regions, the largest individual impact on the total systematic uncertainty is given by the conservative 30% estimate on the c-tagging scale factor uncertainties.

In SRA, other important uncertainties include the Z+c$\bar{\text{c}}$b$\bar{\text{b}}$ uncertainty, uncertainties on $\mu_{\text{Z,AB}}$, the single-top DS and DR schemes as well as jet energy scale (JES) and large-radius jet mass scale (JMS) uncertainties.

The most dominant systematic uncertainties after the c-tagging scale factor uncertainties in SRB0 comprise jet energy resolution (JER) and top-tagging uncertainties, while the Z+c$\bar{\text{c}}$b$\bar{\text{b}}$ uncertainty plays a less important role in SRB0 due to the relatively small contribution of the Z+c$\bar{\text{c}}$b$\bar{\text{b}}$ background component in this region. Also, uncertainties on the background estimates arising from limited MC statistics, denoted by γ_stat_SRB0_cuts in Table 9.28, have a large impact, especially in the SRB0-400 bin.

In addition to the c-tagging scale factor uncertainties, important uncertainties in SRB1 include uncertainties due to limited MC statistics in the considered regions, jet kinematics uncertainties such as α_JET_GroupedNP_2, as well as $E_{\text{T}}^{\text{miss}}$, JER and JES uncertainties. Due to the very similar composition of Z+jets events between CRs and SRs, Z+jets theory uncertainties as well as PDF uncertainties on the transfer factor cancel to a great extent and are mostly found to contribute at a level of less than 2%.

Given the very large dominance of the c-tagging scale factor uncertainties, the introduction of the additional Z+c$\bar{\text{c}}$b$\bar{\text{b}}$ uncertainty discussed in Section 9.9.2 has only a minor impact on the overall systematic uncertainty in the SRs. In fact, in comparison to a background-only fit configuration without this additional uncertainty, the inclusion of the extra Z+c$\bar{\text{c}}$b$\bar{\text{b}}$ uncertainty increases the overall post-fit systematic uncertainty in SRA-450 and SRA-575 only by 0.70% and 0.59%, respectively, while the overall systematic uncertainty in SRB0-100, SRB0-150 and SRB0-400 increases by 0.26%, 0.46% and 0.52%, respectively. The largest impact on the overall systematic uncertainty can be seen in the SRB1-300 bin with

an increment of 1.49%, while the extra Z+c$\bar{c}$b$\bar{b}$ uncertainty increases the total systematic uncertainty in SRB1-100, SRB1-150 and SRB1-500 by 0.77%, 0.97% and 0.13%, respectively.
In SRC, the Z+c$\bar{c}$b$\bar{b}$ uncertainty plays only a negligible role in general, as Z+jets background processes are only sub-dominant.
Although high systematic uncertainties of more than 50% can be seen in several SR bins, the total uncertainty in most bins of the SRA and SRB regions are still dominated by statistical uncertainties.
A summary of all post-fit MC predictions and observed data yields in the unblinded SRs of this analysis is given in Fig. 9.33, where the aforementioned multiple minor excesses in regions SRA, SRB0 and SRB1 and the good agreement between post-fit MC prediction and data in SRC can be seen again.

Since no statistically significant excess in data could be observed, model-independent upper limits on the visible cross section of new physics beyond the Standard Model as well as model-dependent exclusion limits will be derived in the following. Given the slight data overfluctuations observed in multiple SR bins, the observed limits will, however, be weaker than the expected ones.

Region	SRA-450	SRA-575
Total background expectation	4.07	2.88
Total statistical uncertainty	±2.02	±1.70
Total systematic uncertainty	±1.95 [48.02%]	±1.46 [50.68%]
α_ctag_weight	±1.22 [**30.0%**]	±0.86 [**30.0%**]
α_Zccbb	±0.52 [**12.8%**]	±0.22 [7.7%]
μ_Z_AB	±0.37 [**9.1%**]	±0.23 [8.0%]
α_st_theory_PS	±0.32 [8.0%]	±0.04 [1.4%]
α_JES_R10_3	±0.32 [7.8%]	±0.10 [3.3%]
γ_stat_SRA_cuts	±0.32 [7.8%]	±0.26 [8.9%]
α_JES_R10_1	±0.32 [7.8%]	±0.11 [3.8%]
α_JES_R10_4	±0.31 [7.6%]	±0.10 [3.4%]
α_JES_R10_6restTerm	±0.31 [7.6%]	±0.10 [3.4%]
α_JES_R10_5	±0.30 [7.5%]	±0.10 [3.3%]
α_JES_R10_Modelling1	±0.30 [7.3%]	±0.10 [3.4%]
α_JES_R10_Mixed1	±0.29 [7.2%]	±0.10 [3.4%]
α_JES_R10_2	±0.29 [7.2%]	±0.09 [3.0%]
α_JMS_Rtrk_InterDiff	±0.28 [6.9%]	±0.30 [10.4%]
α_JMS_Rtrk_Stat	±0.28 [6.9%]	±0.30 [10.4%]
α_JMS_Rtrk_Gen_InterDiff	±0.26 [6.5%]	±0.27 [9.4%]
α_JMS_FF_Shape	±0.25 [6.0%]	±0.32 [**10.9%**]
α_JMS_Rtrk_Gen	±0.24 [6.0%]	±0.31 [10.9%]
α_JMS_FF_Stat	±0.23 [5.7%]	±0.30 [10.4%]
μ_st_AB1	±0.22 [5.4%]	±0.25 [8.8%]
α_JMS_FF_PartonShower	±0.22 [5.3%]	±0.30 [10.4%]
α_st_theory_DS	±0.21 [5.2%]	±0.52 [**18.0%**]
α_st_theory_GEN	±0.20 [4.8%]	±0.15 [5.1%]
α_JMS_Rtrk_Tracking	±0.18 [4.4%]	±0.14 [5.0%]
α_JER_EffectiveNP_4	±0.18 [4.4%]	±0.05 [1.7%]
α_W_theory_renorm	±0.16 [4.0%]	±0.09 [3.3%]
α_JER_EffectiveNP_3	±0.14 [3.5%]	±0.10 [3.4%]
α_JET_GroupedNP_1	±0.14 [3.5%]	±0.00 [0.08%]
α_JER_EffectiveNP_1	±0.13 [3.3%]	±0.06 [2.1%]
α_JET_GroupedNP_2	±0.13 [3.2%]	±0.00 [0.01%]
α_JMS_Topology_QCD	±0.12 [2.9%]	±0.01 [0.39%]
α_JMS_FF_LargerSample	±0.11 [2.8%]	±0.27 [9.3%]
α_JER_EffectiveNP_2	±0.11 [2.7%]	±0.04 [1.5%]
α_JER_dijet_R10_stat	±0.09 [2.1%]	±0.04 [1.2%]
α_JER_dijet_R10_jesEffNP3	±0.08 [2.1%]	±0.05 [1.7%]
α_JER_dijet_R10_jesEtaIntMod	±0.08 [2.1%]	±0.05 [1.7%]
α_JER_dijet_R10_jesFlavResp	±0.08 [2.1%]	±0.05 [1.7%]
α_JER_dijet_R10_jesFlavComp	±0.08 [2.0%]	±0.05 [1.7%]
α_JER_dijet_R10_selection	±0.08 [2.0%]	±0.05 [1.8%]

Table 9.27: Breakdown of the dominant systematic uncertainties in the SRA bins after the background-only fit. Individual uncertainties can be correlated and do not necessarily add up in quadrature. The three most dominant systematic uncertainties in each region are shown in bold.

Region	SRB0-100	SRB0-150	SRB0-400
Total background expectation	7.33	11.52	2.55
Total statistical uncertainty	±2.71	±3.39	±1.60
Total systematic uncertainty	±4.53 [61.82%]	±5.15 [44.72%]	±1.32 [51.67%]
α_ctag_weight	±2.19 [**30.0%**]	±3.46 [**30.0%**]	±0.77 [**30.0%**]
α_JER_EffectiveNP_6	±1.11 [**15.1%**]	±1.40 [**12.1%**]	±0.03 [1.00%]
α_JER_EffectiveNP_7restTerm	±1.03 [**14.0%**]	±1.44 [**12.5%**]	±0.06 [2.5%]
α_JER_EffectiveNP_5	±1.00 [13.6%]	±0.68 [5.9%]	±0.06 [2.3%]
γ_stat_SRB0_cuts	±0.92 [12.6%]	±0.92 [8.0%]	±0.41 [**16.2%**]
α_JER_EffectiveNP_3	±0.71 [9.7%]	±0.85 [7.4%]	±0.04 [1.8%]
α_JES_R10_2	±0.69 [9.4%]	±0.18 [1.6%]	±0.16 [6.4%]
α_JER_dijet_R10_closure	±0.69 [9.3%]	±0.40 [3.5%]	±0.14 [5.6%]
α_JER_EffectiveNP_4	±0.68 [9.3%]	±0.50 [4.4%]	±0.03 [1.4%]
α_JES_R10_4	±0.67 [9.2%]	±0.17 [1.5%]	±0.18 [6.9%]
α_JER_dijet_R10_jesEffNP1	±0.65 [8.9%]	±0.55 [4.7%]	±0.12 [4.6%]
α_TopTag_Radiation	±0.65 [8.8%]	±0.55 [4.8%]	±0.04 [1.7%]
α_TopTag_Hadronisation	±0.65 [8.8%]	±0.55 [4.8%]	±0.04 [1.7%]
α_TopTag_Dijet_Modelling	±0.65 [8.8%]	±0.55 [4.8%]	±0.04 [1.7%]
α_TopTag_MatrixElement	±0.65 [8.8%]	±0.55 [4.8%]	±0.04 [1.7%]
α_TopTag_BGSF_Dijet_Stat	±0.65 [8.8%]	±0.55 [4.8%]	±0.04 [1.7%]
α_TopTag_γjet_Modelling	±0.65 [8.8%]	±0.55 [4.8%]	±0.04 [1.7%]
α_JES_R10_5	±0.65 [8.8%]	±0.12 [1.1%]	±0.18 [6.9%]
α_JES_R10_Mixed1	±0.65 [8.8%]	±0.14 [1.2%]	±0.18 [6.9%]
α_JES_R10_6restTerm	±0.65 [8.8%]	±0.13 [1.1%]	±0.18 [6.9%]
α_JES_R10_Modelling1	±0.65 [8.8%]	±0.14 [1.3%]	±0.18 [6.9%]
μ_Z_AB	±0.61 [8.3%]	±1.03 [9.0%]	±0.33 [13.0%]
α_JES_R10_1	±0.57 [7.7%]	±0.09 [0.74%]	±0.18 [7.1%]
α_JES_R10_3	±0.56 [7.7%]	±0.09 [0.80%]	±0.18 [7.1%]
α_JER_dijet_R10_jesEtaIntMod	±0.53 [7.3%]	±0.12 [1.1%]	±0.07 [2.9%]
α_JER_dijet_R10_jesEffNP3	±0.53 [7.2%]	±0.14 [1.2%]	±0.05 [1.9%]
α_JER_dijet_R10_jesFlavResp	±0.52 [7.1%]	±0.07 [0.63%]	±0.11 [4.3%]
α_JER_dijet_R10_selection	±0.52 [7.1%]	±0.01 [0.06%]	±0.11 [4.3%]
α_st_theory_DS	±0.50 [6.8%]	±0.49 [4.2%]	±0.11 [4.2%]
α_JER_dijet_R10_mcGen	±0.49 [6.7%]	±0.13 [1.1%]	±0.10 [3.8%]
α_JMS_FF_LargerSample	±0.49 [6.7%]	±0.65 [5.6%]	±0.12 [4.9%]
α_JER_dijet_R10_jesFlavComp	±0.49 [6.6%]	±0.03 [0.25%]	±0.09 [3.4%]
α_JER_EffectiveNP_2	±0.48 [6.6%]	±0.47 [4.1%]	±0.35 [**13.6%**]
α_JER_dijet_R10_jesEffNP4	±0.47 [6.5%]	±0.11 [0.96%]	±0.08 [3.1%]
α_JET_Flavor_Response	±0.46 [6.3%]	±0.45 [3.9%]	±0.07 [2.7%]
α_W_theory_renorm	±0.46 [6.3%]	±0.76 [6.6%]	±0.09 [3.7%]
α_JET_GroupedNP_2	±0.46 [6.2%]	±0.29 [2.6%]	±0.13 [5.2%]
α_JER_dijet_R10_stat	±0.44 [6.0%]	±0.00 [0.00%]	±0.09 [3.6%]
α_Zccbb	±0.42 [5.7%]	±1.05 [9.1%]	±0.25 [10.0%]

Table 9.28: Breakdown of the dominant systematic uncertainties in the SRB0 bins after the background-only fit. Individual uncertainties can be correlated and do not necessarily add up in quadrature. The three most dominant systematic uncertainties in each region are shown in bold.

Region	SRB1-100	SRB1-150	SRB1-300	SRB1-500
Total background expectation	3.49	4.78	6.25	2.22
Total statistical uncertainty	±1.87	±2.19	±2.50	±1.49
Total systematic uncertainty	±1.79 [51.20%]	±1.96 [40.92%]	±2.80 [44.84%]	±1.21 [54.30%]
α_ctag_weight	±1.05 [**30.0%**]	±1.43 [**30.0%**]	±1.88 [**30.0%**]	±0.67 [**30.0%**]
γ_stat_SRB1_cuts	±0.45 [**12.9%**]	±0.53 [**11.1%**]	±0.64 [**10.3%**]	±0.41 [**18.5%**]
α_JET_GroupedNP_2	±0.44 [**12.6%**]	±0.35 [7.3%]	±0.48 [7.7%]	±0.05 [2.1%]
α_MET_SoftTrk_ResoPerp	±0.42 [12.1%]	±0.05 [1.1%]	±0.07 [1.1%]	±0.16 [7.2%]
α_Zccbb	±0.39 [11.3%]	±0.46 [**9.7%**]	±1.12 [**17.9%**]	±0.08 [3.8%]
α_MET_SoftTrk_ResoPara	±0.35 [10.1%]	±0.09 [1.9%]	±0.19 [3.0%]	±0.10 [4.4%]
α_W_theory_renorm	±0.31 [8.9%]	±0.16 [3.4%]	±0.36 [5.8%]	±0.11 [4.8%]
α_JET_Flavor_Response	±0.29 [8.2%]	±0.32 [6.8%]	±0.23 [3.7%]	±0.02 [0.93%]
α_JER_MC16	±0.28 [7.9%]	±0.11 [2.4%]	±0.04 [0.69%]	±0.10 [4.6%]
α_JES_R10_6restTerm	±0.26 [7.6%]	±0.09 [1.9%]	±0.03 [0.50%]	±0.06 [2.7%]
α_JES_R10_4	±0.26 [7.6%]	±0.09 [1.8%]	±0.03 [0.52%]	±0.06 [2.7%]
α_JES_R10_2	±0.26 [7.4%]	±0.07 [1.4%]	±0.03 [0.40%]	±0.06 [2.7%]
α_JER_R10_MC16	±0.26 [7.4%]	±0.13 [2.7%]	±0.29 [4.7%]	±0.06 [2.6%]
α_JES_R10_Modelling1	±0.25 [7.3%]	±0.10 [2.2%]	±0.03 [0.47%]	±0.06 [2.8%]
α_JES_R10_3	±0.25 [7.2%]	±0.09 [1.9%]	±0.03 [0.46%]	±0.07 [3.1%]
α_JES_R10_1	±0.25 [7.2%]	±0.09 [1.9%]	±0.02 [0.36%]	±0.05 [2.2%]
α_JES_R10_5	±0.25 [7.2%]	±0.09 [1.9%]	±0.03 [0.49%]	±0.05 [2.2%]
α_JES_R10_Mixed1	±0.25 [7.1%]	±0.09 [1.9%]	±0.03 [0.50%]	±0.06 [2.7%]
α_JER_AllOthers	±0.25 [7.1%]	±0.17 [3.5%]	±0.29 [4.7%]	±0.04 [1.9%]
α_JES_R10_EtaInterCalib	±0.23 [6.6%]	±0.16 [3.3%]	±0.16 [2.6%]	±0.07 [3.2%]
α_JET_GroupedNP_1	±0.22 [6.3%]	±0.11 [2.3%]	±0.55 [8.8%]	±0.05 [2.1%]
α_JMS_Rtrk_Gen	±0.22 [6.2%]	±0.08 [1.7%]	±0.20 [3.3%]	±0.07 [3.2%]
α_JMS_FF_Stat	±0.22 [6.2%]	±0.07 [1.4%]	±0.20 [3.1%]	±0.08 [3.5%]
α_JMS_FF_PartonShower	±0.22 [6.2%]	±0.07 [1.4%]	±0.21 [3.4%]	±0.08 [3.5%]
α_JMS_FF_Shape	±0.22 [6.2%]	±0.08 [1.6%]	±0.20 [3.3%]	±0.07 [3.2%]
μ_st_AB1	±0.21 [6.0%]	±0.28 [5.9%]	±0.43 [7.0%]	±0.07 [3.0%]
α_JMS_Rtrk_Stat	±0.18 [5.1%]	±0.07 [1.4%]	±0.20 [3.1%]	±0.09 [3.9%]
α_JMS_Rtrk_InterDiff	±0.18 [5.1%]	±0.07 [1.4%]	±0.20 [3.1%]	±0.09 [3.9%]
α_JMS_Rtrk_Tracking	±0.17 [5.0%]	±0.04 [0.74%]	±0.14 [2.2%]	±0.09 [3.9%]
α_st_theory_PS	±0.17 [4.9%]	±0.29 [6.0%]	±0.13 [2.0%]	±0.06 [2.9%]
α_JMS_Rtrk_Gen_InterDiff	±0.16 [4.6%]	±0.05 [1.1%]	±0.20 [3.2%]	±0.01 [0.60%]
α_JER_EffectiveNP_6	±0.15 [4.3%]	±0.19 [3.9%]	±0.07 [1.1%]	±0.04 [1.8%]
μ_Z_AB	±0.14 [4.1%]	±0.37 [7.8%]	±0.60 [9.7%]	±0.26 [**11.7%**]
α_JER_EffectiveNP_7restTerm	±0.11 [3.1%]	±0.31 [6.5%]	±0.11 [1.8%]	±0.02 [1.1%]
α_JER_dijet_R10_jesEffNP1	±0.10 [3.0%]	±0.10 [2.1%]	±0.39 [6.2%]	±0.00 [0.12%]
α_JER_dijet_R10_closure	±0.10 [3.0%]	±0.11 [2.3%]	±0.38 [6.1%]	±0.01 [0.33%]
α_JMS_FF_LargerSample	±0.10 [2.9%]	±0.15 [3.1%]	±0.17 [2.8%]	±0.05 [2.3%]
α_JER_EffectiveNP_1	±0.09 [2.6%]	±0.02 [0.42%]	±0.51 [8.2%]	±0.16 [7.3%]
α_FT_EFF_L_1_systematics	±0.07 [2.0%]	±0.01 [0.22%]	±0.03 [0.44%]	±0.06 [2.7%]
α_ttbar_theory_PS	±0.07 [2.0%]	±0.02 [0.39%]	±0.22 [3.5%]	±0.07 [3.3%]

Table 9.29: Breakdown of the dominant systematic uncertainties in the SRB1 bins after the background-only fit. Individual uncertainties can be correlated and do not necessarily add up in quadrature. The three most dominant systematic uncertainties in each region are shown in bold.

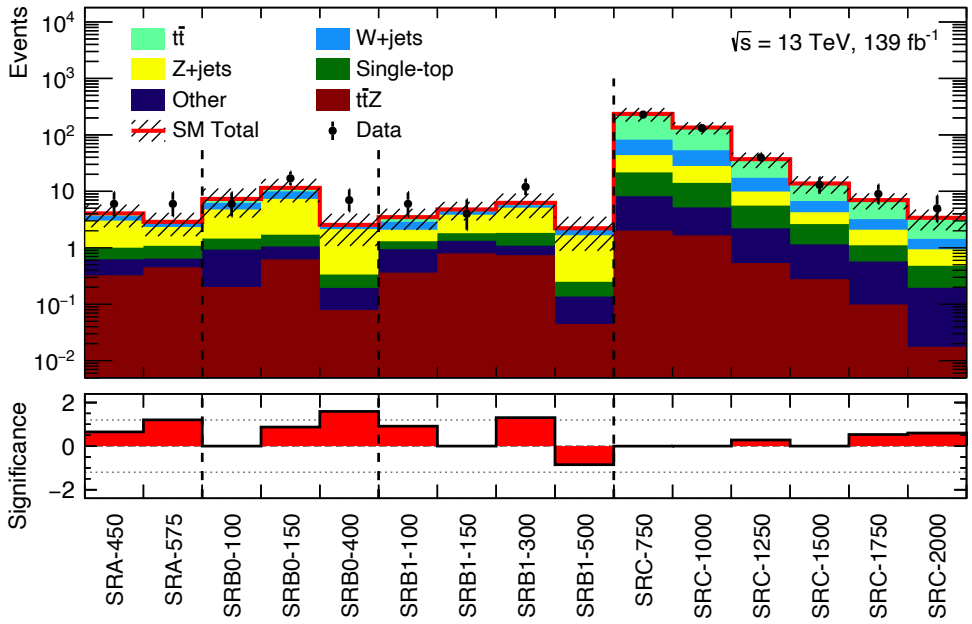


Figure 9.33: Summary of all MC predicted backgrounds and observed data events after the background-only fit (top panel), together with the resulting significances (bottom panel) in all SRs of this analysis.

9.10.2 Interpretation

Model-independent upper limits as well as model-dependent exclusion limits have been derived and will be presented in the following.

Model-independent Upper Limits

The visible cross section σ_{vis} of an arbitrary BSM process, that could potentially enter this analysis' SRs, is defined as

$$\sigma_{\text{vis}} = \sigma_{\text{BSM}} \cdot \epsilon_{\text{sel}} \cdot \epsilon_{\text{acc}}, \tag{9.11}$$

where σ_{BSM} denotes the BSM process production cross section and ϵ_{sel} as well as ϵ_{acc} the analysis selection and acceptance efficiencies, respectively.
Using the model-independent fit configuration discussed in Section 8.3 with the inclusive regions SRA, SRB0, SRB1 as well as SRC-750, SRC-1000 and SRC-1250, upper limits on σ_{vis} can be derived by letting the fit determine the signal strength parameter μ_{sig} and the number of signal events of an arbitrary signal entering the individual SRs for which the corresponding CL_s value drops below 0.05. The upper limit on the number of observed signal events S^{95}_{obs} can then be divided by the integrated luminosity of 139 fb^{-1} to determine the upper limit on σ_{vis}, denoted by $\langle\sigma_{\text{vis}}\rangle^{95}_{\text{obs}}$.
Table 9.30 lists the observed upper limits on the visible cross section together with the observed and expected upper limits on the number of signal events as well as the discovery p-value p_0 with its associated significance Z for the inclusive discovery regions SRA, SRB0,

SRB1, SRC-750, SRC-1000 and SRC-1250.
Observed upper limits on the visible cross section of 0.09 fb, 0.14 fb, 0.11 fb, 0.12 fb, 0.12 fb and 0.24 fb are obtained for the regions SRA, SRB0, SRB1, SRC-750, SRC-1000 and SRC-1250, respectively, and the background hypothesis cannot be rejected, as all significance values are well below 2σ.

Region	$\langle\sigma_{\text{vis}}\rangle^{95}_{\text{obs}}$[fb]	S^{95}_{obs}	S^{95}_{exp}	p_0	Z
SRA	0.09	12.7	$7.7^{+4.4}_{-3.6}$	0.09	1.33
SRB0	0.14	19.9	$12.5^{+6.6}_{-4.3}$	0.11	1.21
SRB1	0.11	15.4	$10.3^{+4.7}_{-3.4}$	0.15	1.04
SRC-750	0.12	17.2	$16.2^{+5.7}_{-3.5}$	0.34	0.40
SRC-1000	0.12	17.1	$17.1^{+5.1}_{-3.7}$	0.50	0.00
SRC-1250	0.24	32.8	$28.5^{+8.0}_{-6.8}$	0.49	0.03

Table 9.30: Model-independent upper limits on the visible cross section and the number of signal events in all discovery signal regions, together with the respective discovery p-values and significances.

Model-dependent Exclusion Limits

In addition to model-independent upper limits, model-dependent limits for each signal point in the signal grid can be derived using combined exclusion fits of all exclusion SR bins, which were specifically designed to be orthogonal to one another and hence allow for a statistical combination.
Fig. 9.34 presents the expected and observed exclusion limits in the $m(\tilde{t}_1)$–$m(\tilde{\chi}^0_1)$ plane at $\text{CL}_s = 0.05$, excluding signal scenarios inside the observed contours at a 95% CL. The $\pm1\sigma$ uncertainties on the expected limits as well as – for reference – also the $\pm2\sigma$ uncertainties are shown in addition.
While the expected limits reach up to more than 1 TeV in stop mass for small neutralino masses and up to $m(\tilde{t}_1) \approx 900\,\text{GeV}$ for $m(\tilde{\chi}^0_1) \approx 400\,\text{GeV}$, the observed limits are reduced due to the slight overfluctuations of data seen in multiple SR bins. For low neutralino masses, stop masses below $m(\tilde{t}_1) \approx 880\,\text{GeV}$ as well as $m(\tilde{t}_1) \lesssim 760\,\text{GeV}$ for $m(\tilde{\chi}^0_1) \approx 300\,\text{GeV}$ can be excluded at a 95% CL for signal scenarios of flavor-violating supersymmetry with branching ratios of $\text{BR}(\tilde{t}_1 \to \text{t}\tilde{\chi}^0_1) = \text{BR}(\tilde{t}_1 \to \text{c}\tilde{\chi}^0_1) = 50\%$. Overall, the observed limit is found within the 2σ uncertainty band of the expected one.
By reweighting the simulated MC signal samples according to their respective stop-pair decay types ($\text{t}\bar{\text{c}}/\bar{\text{t}}\text{c} + \tilde{\chi}^0_1\tilde{\chi}^0_1$, $\text{t}\bar{\text{t}} + \tilde{\chi}^0_1\tilde{\chi}^0_1$ or $\text{c}\bar{\text{c}} + \tilde{\chi}^0_1\tilde{\chi}^0_1$), the model-dependent exclusion fits can also be reinterpreted for branching ratios different from $\text{BR}(\tilde{t}_1 \to \text{t}\tilde{\chi}^0_1) = \text{BR}(\tilde{t}_1 \to \text{c}\tilde{\chi}^0_1) = 50\%$, allowing to set limits in the $m(\tilde{t}_1)$–$\text{BR}(\tilde{t}_1 \to \text{t}\tilde{\chi}^0_1)$ plane. Fig. 9.35 shows the resulting expected and observed exclusion contours in the $m(\tilde{t}_1)$–$\text{BR}(\tilde{t}_1 \to \text{t}\tilde{\chi}^0_1)$ plane for neutralino masses of $m(\tilde{\chi}^0_1) = 1\,\text{GeV}$ and $m(\tilde{\chi}^0_1) = 200\,\text{GeV}$.

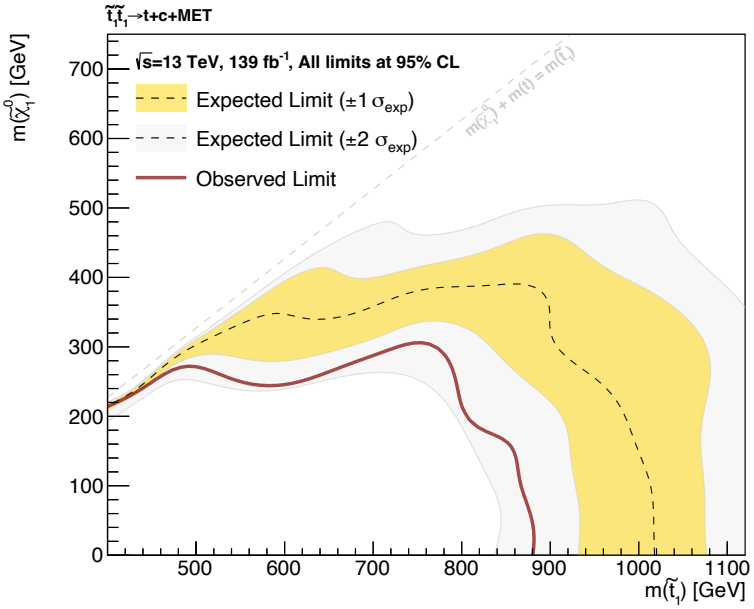
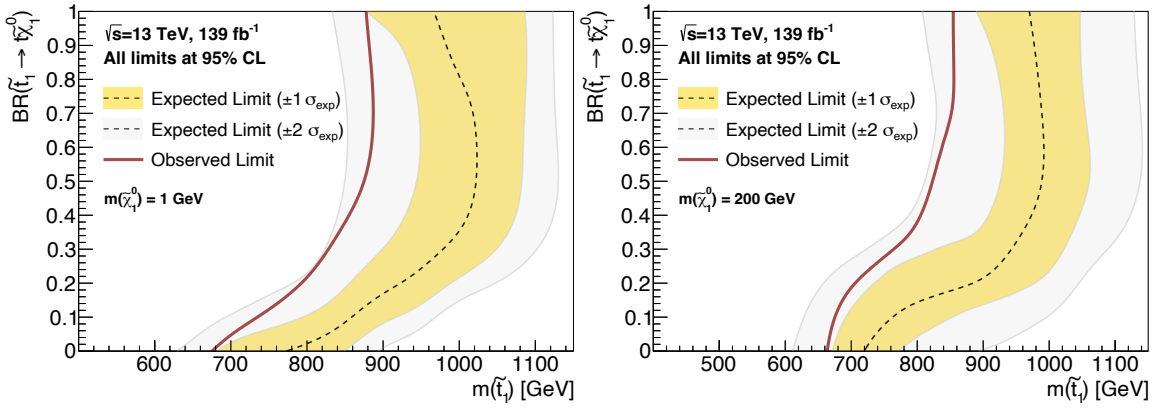


Figure 9.34: Model-dependent exclusion contours in the $m(\tilde{t}_1)$–$m(\tilde{\chi}_1^0)$ plane at 95% CL. The black dashed line represents the expected limit with its $\pm1\sigma$ and $\pm2\sigma$ uncertainties given by the yellow and light gray bands, respectively, while the observed limit is given by the red solid line.



(a) (b)

Figure 9.35: Model-dependent exclusion contours in the $m(\tilde{t}_1)$–$\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0)$ plane at 95% CL for (a) $m(\tilde{\chi}_1^0) = 1\,\mathrm{GeV}$ and (b) $m(\tilde{\chi}_1^0) = 200\,\mathrm{GeV}$. The nominal $\mathrm{tc} + E_\mathrm{T}^\mathrm{miss}$ final state of this analysis is given by branching ratio values of $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) = 0.5$, whereas $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) = 0$ and $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) = 1$ imply pure $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$ and $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$ final states, respectively. The black dashed line represents the expected limit with its $\pm1\sigma$ and $\pm2\sigma$ uncertainties given by the yellow and light gray bands, respectively, while the observed limit is shown by the red solid line.

For both neutralino mass scenarios, expected limits for $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) > 0.5$ extend up to $950 - 1000\,\mathrm{GeV}$ in stop mass, while the exclusion power is weakened for smaller branching ratios, and thus more $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$-like final states. This behavior can be understood by the applied b- and top-tag requirements in most of the SRs which are difficult to fulfill for such final states. Especially SRB0, requiring no top-tags and still being sensitive to larger mass splittings, enables expected exclusion limits of up to 780 GeV for $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) \approx 0.0$ and $m(\tilde{\chi}_1^0) = 1\,\mathrm{GeV}$ while relying on ISR-initiated b-jets or fake b-tagged c-jets. Again, the observed limits are weaker than the expected ones due to the slight data overfluctuations seen in several SR bins, excluding stop masses up to 880 GeV for $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) > 0.5$ and up to 680 GeV for $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) \approx 0.0$, and are still found within the 2σ uncertainty bands of the expected exclusion contours.

9.10.3 Discussion

Although the slight data overfluctuations observed in several SR bins reduce the exclusion capabilities for new physics models, this first-of-its-kind analysis in ATLAS sets the most stringent limits for flavor-violating supersymmetry in the targeted parameter space at the time of writing. The observed limits strongly improve on the limits shown in Fig. 2.6, that were derived in Ref. [41] by reinterpreting the existing ATLAS $36.1\,\mathrm{fb}^{-1}$ $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$ [39] and ATLAS $36.1\,\mathrm{fb}^{-1}$ $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$ [40] searches, while the analysis' expected limits with a dataset of $139\,\mathrm{fb}^{-1}$ are found to coincide with the $300\,\mathrm{fb}^{-1}$ prospects presented in Ref. [41] for maximal mixing scenarios. For more $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$- and $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$-like signal scenarios, the expected limits of this analysis even exceed the HL-LHC prospects of a $\mathrm{tc} + E_\mathrm{T}^\mathrm{miss}$ analysis presented in Ref. [41], while the data overfluctuations in the SRs reduce the observed limits to stop masses of about 880 GeV and 680 GeV for pure $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$ and $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$ final states, respectively, in case of $m(\tilde{\chi}_1^0) = 1\,\mathrm{GeV}$. As all individual overfluctuations are observed in statistically independent signal region bins, the overall observed exclusion limits in both the $m(\tilde{t}_1)$–$m(\tilde{\chi}_1^0)$ plane as well as the $m(\tilde{t}_1)$–$\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0)$ plane are found just within a 2σ deviation from the expected ones.

The true origin of the excesses remains unclear and could be explained by pure statistical fluctuations, an underestimated background process in the considered regions of phase space or, in the best case, a first possible hint for a new physics process beyond the Standard Model. Assuming a branching ratio of $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) = \mathrm{BR}(\tilde{t}_1 \to \mathrm{c}\tilde{\chi}_1^0) = 50\%$, analyses searching for new physics in $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$ or $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$ final states should also be sensitive to these signal models, however, with a reduced sensitivity due to only 25% of the events containing pure $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$ or $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$ final states. While, for example, the results of the ATLAS Run 2 $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$ 1-lepton search [179] also show a small (but statistically not significant) excess of 1.9σ in a signal region sensitive to large $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$ and also the CMS Run 2 $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$ 1-lepton search [182] observes multiple overfluctuations of about 3σ in some of their high-$E_\mathrm{T}^\mathrm{miss}$ regions, no comparable excesses are seen in e.g. the ATLAS Run 2 $\mathrm{t\bar{t}} + E_\mathrm{T}^\mathrm{miss}$ 2-lepton search [183]. The ATLAS Run 2 $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$ analysis is still blinded at the time of writing and has no published results yet. A statistical combination of both the ATLAS Run 2 $\mathrm{tc} + E_\mathrm{T}^\mathrm{miss}$ analysis presented in this thesis and the ATLAS Run 2 $\mathrm{c\bar{c}} + E_\mathrm{T}^\mathrm{miss}$

analysis will be possible in the future, as both analyses were specifically designed for such an eventual combination[4].

At the time of writing, this analysis makes use of a conservative 30% estimate on the c-tagging scale factor uncertainty, which was found to be by far the largest individual contributor to the total systematic uncertainties in all SRs. Once the real c-tagging scale factor uncertainties are available and included in the analysis, the total systematic uncertainties are expected to be reduced significantly, making the $Z+c\bar{c}b\bar{b}$ uncertainty the dominant one in several SR bins. In addition, while several SR bins are currently dominated by high systematic uncertainties, most SR bins are expected to be fully dominated by statistics when reducing the large c-tagging scale factor uncertainty.

The largest impact on the available statistics in both SRs and CRs is given by the employed heavy-flavor tagging methods, in particular the charm-tagging method with only 20% efficiency. Given the low charm-tagging efficiency, a top-tagging method with a very high mistag rate has to be used to retain enough statistics, resulting in Z+jets events with fake top-jets as the dominant background in SRA and SRB1, where at least one top-tagged jet is required. Hence, improvements on the available charm-tagging methods would significantly enhance the analysis' performance, as more statistics in both SRs and CRs could be retained while important backgrounds in several SRs could be drastically reduced by allowing to employ more efficient top-tagging methods.

In order to allow for a better understanding of the nature of the several small excesses, a next iteration of this analysis could largely profit from the resulting lower systematic and statistical uncertainties as well as from improved background estimation strategies. Here, also data driven methods for Z+jets modeling or an additional dedicated CR (similar to the region defined in Appendix B.7.2) for the $Z+c\bar{c}b\bar{b}$ component, whose uncertainty will be one of the most dominant ones at this point, could help to further improve the sensitivity of the analysis.

[4] A straightforward combination with the ATLAS Run 2 $t\bar{t} + E_{\mathrm{T}}^{\mathrm{miss}}$ analyses will unfortunately not be possible, as these older analyses make use of different object definitions, tagging methods and software frameworks.

Chapter 10

Summary and Outlook

The vertexing work presented in this thesis reports on the development of novel primary vertexing algorithms as well as a new, highly performant and thread-safe primary vertex reconstruction software suite implemented in the ACTS software toolkit in preparation for the upcoming challenging computational conditions for HL-LHC and beyond.
New algorithmic developments include the introduction of a generalized approach for time-dependent track linearization as well as the development and implementation of the new (Adaptive) Grid Seeder algorithm. This innovative approach to primary vertex seed finding was specifically developed to overcome the severe algorithmic limitations seen for previous primary vertex seed finder algorithms in high-luminosity environments.
During the development of the ACTS vertexing software suite, special care was taken to guarantee maximum flexibility and robustness for an experiment-independent utilization of the implemented algorithms while simultaneously providing an excellent computational performance as well as an underlying four-dimensional event data model to incorporate timing information into vertex reconstruction for the first time.
In-depth CPU performance optimization studies significantly improved the software's computational performance, reducing the required average primary vertex reconstruction time per event of the Adaptive Multi-Vertex Finder algorithm by more than a factor of two, without introducing any algorithmic changes to the behavior of the vertexing components.
The ACTS vertexing software suite was subsequently integrated, tested and validated with respect to its physics and computational performance in the lightweight ACTS-Examples framework as well as in the ATLAS software framework Athena. The application of both major primary vertex finding approaches implemented in ACTS, the ACTS AMVF and the ACTS IVF algorithm, leads to an excellent CPU performance improvement of the primary vertexing reconstruction in the ATLAS software framework, while the resulting physics performance of the employed algorithms could be seen to be entirely identical compared to their respective previous ATLAS implementations.
Furthermore, the application of the newly developed Grid Seeder within the ACTS AMVF algorithm in ATLAS leads to an outstanding CPU performance especially in dense, high pile-up environments. It significantly outperforms the previous ATLAS baseline method by reducing the overall primary vertex reconstruction time per event from more than 750 ms (ATLAS) to less than 200 ms (ACTS) for events with high track multiplicities, while, at the

same time, an enhanced physics performance with less contamination from pile-up vertices and thus a better hard-scatter vertex reconstruction quality is observed.
Due to its superb physics and CPU performance, the ACTS primary vertexing software developed in the course of this thesis will be used as the default primary vertex reconstruction tool in ATLAS for LHC Run 3 and marks the first production use of an ACTS software component in an LHC experiment. Furthermore, ATLAS is currently planning to replace its current primary vertex seed finding method with the newly presented Grid Seeder algorithm for the HL-LHC runs due to its excellent physics and CPU performance capabilities in these challenging conditions.
The main goal of the presented vertexing work was to provide modern and state-of-the-art vertex reconstruction software and algorithms with particular focus on maximum computational efficiency. Future developments can build upon this work and further improve the physics performance of the deployed methods in order to guarantee also optimal physics reconstruction efficiency in the upcoming high-luminosity environments. In this context, the application of novel machine learning techniques indicate very promising prospects, as exemplarily shown in Appendix A.12 for the application of a neural network classifier for the detection of pile-up contamination in hard-scatter vertices, and will therefore most likely play a major role in the future of vertex reconstruction methods.

Furthermore, a first-of-its-kind ATLAS search for supersymmetric partners of the top quark in a flavor-violating supersymmetric extension of the Standard Model, using $139\,\mathrm{fb}^{-1}$ of ATLAS proton-proton collision data recorded at a center-of-mass energy of 13 TeV, was presented. Assuming a maximal mixing in the second and third generations of supersymmetric particles with branching ratios of $\mathrm{BR}(\tilde{t}_1 \to \mathrm{t}\tilde{\chi}_1^0) = \mathrm{BR}(\tilde{t}_1 \to \mathrm{c}\tilde{\chi}_1^0) = 50\%$, the search targets an asymmetric 0-lepton $\mathrm{tc} + E_{\mathrm{T}}^{\mathrm{miss}}$ final state and makes use of simultaneous b-, c- and top-tagging techniques.
In order to account for different signal scenarios with different stop and neutralino masses $m(\tilde{t}_1)$ and $m(\tilde{\chi}_1^0)$, multiple signal regions were designed, each one dedicated to target specific values of $\Delta m(\tilde{t}_1, \tilde{\chi}_1^0)$ and thus specific regions in the parameter space of interest.
As no significant excess over the Standard Model prediction was observed, model-independent upper limits on the visible cross section as well as model-dependent exclusion limits have been derived. Here, the occurrence of several small data overfluctuations of up to 1.6σ in multiple statistically independent signal region bins resulted in a reduction of the overall observed exclusion limits with just below 2σ deviation from expected ones. Observed (expected) exclusion limits of up to about 880 GeV (1020 GeV) were derived at 95% CL for masses of a supersymmetric partner of the top quark in a flavor-violating supersymmetric model with maximal mixing in second and third generations, excluding a previously largely unexplored region of supersymmetry parameter space and setting the most stringent limits on these models at the time of writing.
The largest current limitations in the analysis stem from the low c-tagging efficiency and the consequential reduction in statistics, which required the utilization of a top-tagging algorithm with a very low background rejection power and in turn resulted in Z+jets events

with fake top-tagged jets as the dominant background process in several signal regions, as well as the use of the conservative 30% estimate on the c-tagging scale factor uncertainties, given that the real uncertainties are not yet available at the time of writing.
Once the full c-tagging scale factor uncertainties are available and included in the analysis, the total systematic uncertainty is expected to be largely reduced and the total uncertainties in both control as well as signal regions will be dominated by statistics. The analysis will therefore not only profit significantly from future advances in charm tagging methods, but also from the larger datasets that will become available in LHC Run 3 and beyond.
As most enhancements in charm tagging methods heavily rely on novel developments of secondary vertex finding techniques, the developed ACTS vertexing suite will play an important role in such developments in the future by providing a state-of-the-art vertexing code base to lay an important foundation to facilitate efficient testing and deployment of novel secondary vertexing and heavy-flavor tagging methods.
Thus, by improving upon the aforementioned items, a next iteration of this analysis will most likely be able to shed light on the true origin of the slight data excesses observed in the several signal region bins. Depending on their origin, the analysis will then be able to either further extend the current exclusion limits or, in the best case, observe the first indications of physics beyond the Standard Model produced at the Large Hadron Collider.

Appendix A

Appendix A

A.1 ACTS Vertexing Event Data Model

An event data model provides common data structures for intra- and inter-software component communication. The two fundamental building blocks of the ACTS vertexing event data model are the representations of track parameters as well as reconstructed vertex objects, both of which will be discussed in the following.

A.1.1 Track Parameter Representation

ACTS accommodates two different track parameter spaces, so-called *bound* track parameters as well as *free* track parameters. Bound track parameters represent tracks bound to a specific surface, such as a real planar surface describing a sensitive detector element or a virtual perigee surface, used to describe a track at the point of closest approach to a vertex. The bound parameter representation comprises two local position components l_x and l_y, given in the local surface frame, the azimuthal and polar momentum direction angles ϕ and θ, a curvature term $\frac{q}{p}$ with charge q and total momentum p as well as a time component t. The full six-dimensional parametrization can thus be written as

$$\vec{q} = \left(l_x, l_y, \phi, \theta, \frac{q}{p}, t\right). \tag{A.1}$$

In the case of a representation at a perigee surface, the local position components are replaced by the impact parameters (d_0, z_0) and the track parameters are given in a time-augmented perigee frame as already introduced in Section 4.3:

$$\vec{q} = \left(d_0, z_0, \phi, \theta, \frac{q}{p}, t\right) \tag{A.2}$$

Free track parameters, on the other hand, are not bound to any surface and can be seen as a global track representation with the eight-dimensional parameter vector

$$\vec{a} = \left(x, y, z, t, T_x, T_y, T_z, \frac{q}{p}\right), \tag{A.3}$$

where (x, y, z) denote the global track position components at time t with normalized momentum direction components (T_x, T_y, T_z) and curvature $\frac{q}{p}$.
In order to easily represent tracks at a reconstructed vertex and update their parameters with the knowledge of the fitted vertex position, the ACTS vertexing module relies on the bound track parameter representation at a perigee surface and thus mostly uses `Acts::BoundTrackParameters` objects to internally handle the given track collection.
Additionally, linearized track representations are required for most vertex fitter algorithms (see Section 5.3) and thus need to be made available within the vertexing module. `Acts::LinearizedTrack` objects store the linearized state of a track with respect to its associated vertex, i.e. information such as the position Jacobian $\mathbf{D}$, the momentum Jacobian $\mathbf{E}$ and the absolute term $\vec{c}_0$ of the track's parameters linearized at a specific position according to the measurement equation Eq. (5.25).
Sophisticated vertexing algorithms such as the Adaptive Multi-Vertex Finder can associate a single track to several vertices at once during a multi-vertex fit, requiring the software components to simultaneously handle multiple different linearized state objects of a single track, one for each vertex the track is associated to. This motivates the concept of the `Acts::TrackAtVertex` object, encapsulating all important information of a track with respect to a specific vertex. Fig. A.1 shows a class diagram of the `Acts::TrackAtVertex` class, which holds in addition to the linearized track state at the vertex also information about the refitted track parameters with knowledge of the vertex position, vertex fit-related track information such as the track weight, number of degrees of freedom (NDF) and the track χ^2 value, as well as a pointer (i.e. the address in memory) to the original input track parameters.

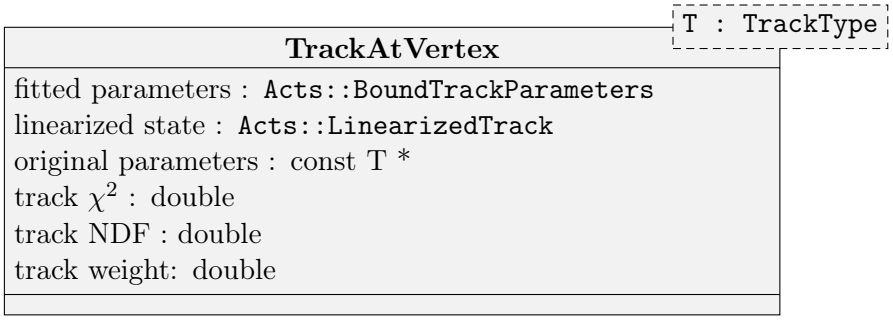


Figure A.1: Class diagram of the `Acts::TrackAtVertex` class. The member variables (left) as well as their respective types (right) are listed. The template parameter `T` specifies the type of the input track parameters, which will be set by the user.

As further discussed in Section A.2, the ACTS vertexing internally always only uses `const` pointers to the original input track objects provided by the user, rather than the track objects themselves. This avoids performance overheads that would otherwise immensely slow down the algorithmic performance of the vertexing tools if, for instance, the content of an entire track collection, such as the current seed track collection, needs to be modified. The usage of pointers allows to keep the original input track objects (that never change throughout the vertexing procedure) in one fixed location in memory and perform

modifications of track collections (i.e. copying, deleting or inserting tracks from specific collections) on a pointer level, making it computationally very inexpensive. The actual object type of the input track parameters `T` is a C++ class template parameter and can be specified by the user, as explained in the next section.
Templates are a key element in the C++ language that allow to write generic classes or functions abstracted from specific types. When instantiating a templated class or function, the types have to be provided as arguments, enabling the compiler to generate specific classes or functions at compilation time without any performance penalties at run time.

A.1.2 User-defined Input Track Type

While the ACTS vertexing EDM is internally based on representing particle track parameters as `Acts::BoundTrackParameters` objects, a user is given the option to represent its input tracks in either this ACTS-specific bound parameter representation (which is the default for employing the vertexing) or to define an own track type to be used for vertex reconstruction. The only requirement for user-defined track types is to implement and provide an `std::function` that unwraps and returns `Acts::BoundTrackParameters` from the track object as exemplarily shown for a very simplistic user-defined track type definition in Listing A.1.
An object of type `UserDefinedTrack` acts as a wrapper of track parameters given in the representation of `Acts::BoundTrackParameters`, which can additionally hold any desired experiment-specific track information. When instantiating the ACTS vertexing algorithm of choice, the corresponding `std::function` object is to be provided to its constructor.
Further information about the actual internal implementation to robustly handle user-defined track types in the vertexing module using template metaprogramming is given in Appendix A.3.

A.1.3 Vertex Representation

The `Acts::Vertex` class, diagrammatically shown in Fig. A.2, represents reconstructed vertices with their respective positions, position covariance matrices, fit quality variables such as the vertex fit χ^2 and number of degrees of freedom as well as all associated tracks that were used to reconstruct and fit the vertex object. The vertex position holds three spatial coordinates as well as a time component in a four-dimensional vector (x, y, z, t), the 4×4 position covariance matrix its associated uncertainties. Note that the entire vertexing EDM was developed in such a way that a temporal component is natively included to allow time-dependent vertex fitting, as will be needed for applications such as the ATLAS HGTD discussed in Section 3.3.2. The associated particle tracks are represented as `Acts::TrackAtVertex<T>` types, making the `Acts::Vertex` class a template type with track type template parameter `T` as well.

```
// Example definition of a user-defined track type
struct UserDefinedTrack {

  /*
     Store any experiment-specific track information here
  */

  // Method to retrieve stored Acts::BoundTrackParameters
  const Acts::BoundTrackParameters&
  params() const {return m_boundParams;}

  // The stored Acts::BoundTrackParameters object
  Acts::BoundTrackParameters m_boundParams;
};

// Corresponding std::function to be provided to the ACTS vertexing
// For any given track object of type UserDefinedTrack, it unwraps
// and returns track parameters in the ACTS bound representation
std::function<Acts::BoundTrackParameters(UserDefinedTrack)>
  extractParameters = [](UserDefinedTrack track)
  { return track.params(); };
```

Listing A.1: Example of a user-specified input track type to the ACTS vertexing with its corresponding required `std::function` definition.

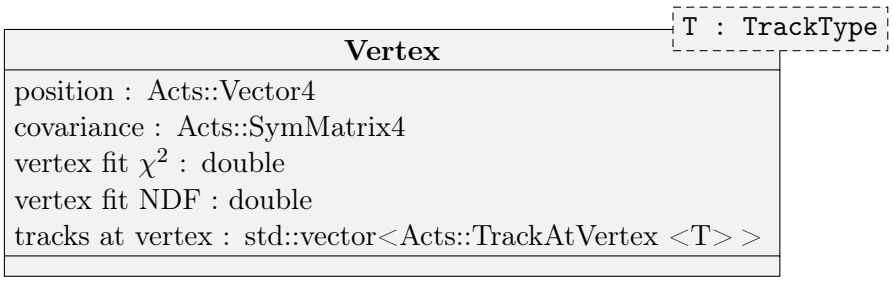


Figure A.2: Class diagram of the `Acts::Vertex` class. The member variables (left) as well as their respective types (right) are listed while public setter and getter methods are not explicitly shown for the sake of conciseness. The template parameter `T` specifies the type of the input track parameters, which will be set by the software's user.

A.2 ACTS Vertexing Template Structure and Public Interfaces

The ACTS vertexing module is designed to provide maximum modularity that allows a flexible and easy interchange of tools and algorithms with similar functionality. In order to avoid any performance overhead at run time induced by virtual inheritance, compile time polymorphism with heavy use of class templates is utilized instead.
The templated type structure of the given algorithms and tools is highly affected by the dependencies of the various vertexing components on each other, as already shown in Fig. 6.1, which will define how a user sets up and and runs the provided vertexing components.
The typical entry point into a vertex reconstruction chain is at the level of the vertex finder algorithm, which internally handles all underlying tools such as the vertex fitter and the vertex seed finder. As illustrated in Fig. A.3, these dependencies are also reflected in the

vertexing module template structure. All ACTS vertex finders depend on the type of the underlying vertex fitter and vertex seed finder to be used. The vertex fitters, in turn, depend on the input track type and the type of the track linearizer, whereas track linearizer types are dependent on the type of the underlying propagation method. The type of the vertex fitter is also used as a template argument for the vertex seed finders, as the fitter type encapsulates all important type information for vertex seed finding such as the track type, linearizer type and propagator type.

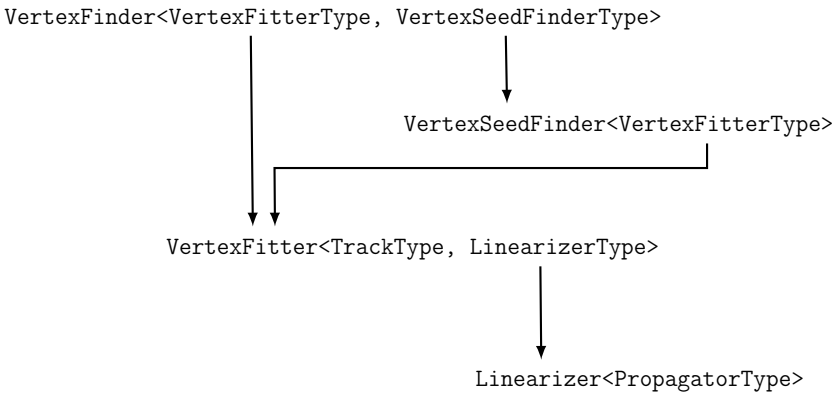


Figure A.3: Template type dependencies of the ACTS vertexing components. Arrows illustrate a dependency on another type, resulting in a deep template structure that allows maximum modularity and interchangeability of various tools and algorithms.

Once a desired type configuration of the vertex finder is chosen (i.e. which underlying vertex fitter, seed finder, track type, linearizer and propagator will be used), an instance of this type can be created and used to find and reconstruct vertices. All vertex finders provide a `find()` method in their public interface that invokes the entire vertex reconstruction chain and returns a set of reconstructed vertices. This method represents the connection element between an experiment's software framework, in which the ACTS vertexing software is used as an external library, and the actual software implementation in ACTS itself and comes with the following signature:

```
Acts::Result<std::vector<Acts::Vertex<TrackType>>> find(
      const std::vector<const TrackType*>& allTracks,
      const Acts::VertexingOptions<TrackType>& vertexingOptions,
      VertexFinderType::State& state) const;
```

Listing A.2: Signature of the ACTS vertex finder `find` methods.

In addition to an input track collection, given as an `std::vector` of `const` pointers to `TrackType` objects, `VertexingOptions` as well as a vertex finder state object have to be provided in the argument list of a vertex finder's `find` method. `VertexingOptions` allow a user to provide additional information to the vertexing algorithms, such as position constraints (e.g. as given by a beam spot) or geometry and magnetic field contextual data. The `State` object ensures that the invoked algorithm itself remains stateless while information

caching for better CPU performance is guaranteed.
The method returns an `Acts::Result` object containing an `std::vector` of all reconstructed vertices if the reconstruction terminated successfully. Otherwise, the type of error that led to an unsuccessful termination of the vertex reconstruction chain together with a descriptive error message can be retrieved from the return object.
For further details on how to set up and run the ACTS vertexing, a dedicated tutorial for developers who would like to integrate the software module in their project was developed and added to the ACTS repository. The tutorial code based on ACTS version 7.0.0 can also be found in the Appendix A.4.

A.3 ACTS Vertexing User-Defined Track Type Implementation

The ACTS vertexing software must provide maximum flexibility for experiment-independent usage while simultaneously guaranteeing best possible computational performance. As described in Section A.1, the developed software allows a highly flexible utilization of user-defined track types as input to the reconstruction algorithms. This section briefly discusses the internal design concept allowing maximum flexibility while ensuring minimum performance overhead.
The implementation of user-defined track types is based on template metaprogramming, specifically on a C++ feature called SFINAE (*Substitution Failure Is Not An Error*), a rule that applies during the resolution of overloads of function templates: when the substitution of a specified type for the template parameter fails (i.e. the specified type is not valid), the specialization is just discarded without resulting in a compilation error. This feature is used in the constructors of all ACTS vertexing classes that are explicitly dependent on the given input track type, as exemplarily shown for the class constructors of the ACTS AMVF in Listing A.3.
The `std::enable_if_t` metafunction in the first constructor leverages SFINAE to conditionally remove the constructor function during compile time if an input track type different from `Acts::BoundTrackParameters` is provided. In this case, the only class constructor available will be the second one, where the `std::function` defined in Listing A.1 with `TrackType` = `UserDefinedTrack` is required as an function argument. Its definition (i.e. the way how to extract `Acts::BoundTrackParameters` from an `UserDefinedTrack` object) is internally stored in a class member `std::function` called `m_extractParameters` that is called every time track parameters of type `Acts::BoundTrackParameters` need to be extracted from a given input track object.
If, on the other hand, `TrackType` = `Acts::BoundTrackParameters`, i.e. the default track type for the ACTS vertexing, is chosen, the first constructor is enabled, allowing to conveniently construct the class without the need of specifying an additional `std::function`. In this case, the class member `m_extractParameters` is just populated with an identity function, such that the input tracks of type `Acts::BoundTrackParameters` given to the

vertexing algorithm can be directly retrieved without any unnecessary performance overhead included.
The above described implementation enables the possibility to consistently use and propagate any user-defined input track type throughout the full vertex reconstruction procedure while ensuring minimal performance overhead. It thus allows for instance to return `Acts::Vertex<TrackType>` objects that can internally store information about the original track objects of type `TrackType`, retaining all possibly required experiment-specific track information together with the ACTS reconstructed vertices.
An example application of this feature is shown in Section 7.3, where the production-ready integration and deployment of the ACTS vertexing software for primary vertex reconstruction in the ATLAS software framework is presented.

```
/*
  Class constructor, only enabled if
  TrackType == BoundTrackParameters
*/
template <typename T = TrackType,
std::enable_if_t<std::is_same<T, Acts::BoundTrackParameters>
  ::value, int> = 0>
Acts::AdaptiveMultiVertexFinder(Config& cfg)
: m_cfg(std::move(cfg))
, m_extractParameters([](T params) { return params; }) {}

/*
  Only class constructor available if user-defined
  TrackType != BoundTrackParameters
*/
Acts::AdaptiveMultiVertexFinder(
    Config& cfg,
    std::function<Acts::BoundTrackParameters(TrackType)> func)
: m_cfg(std::move(cfg))
, m_extractParameters(func){}
```

Listing A.3: The two AMVF class constructors, illustrating how SFINAE is used to conditionally enable or disable constructor functions based on the input track type given to the ACTS vertexing algorithms.

A.4 ACTS Vertexing Tutorial

A short hands-on tutorial to set up and run the ACTS Adaptive Multi-Vertex Finder has been developed and introduced to the ACTS repository, which can be used as a starting point for developers who would like to integrate the ACTS vertexing software in their project. The following tutorial code is a copy available in the ACTS software repository under `acts/docs/howto`, based on ACTS version 7.0.0.

Setting up required tools: stepper and propagator

We need the Acts::Propagator with the Acts::EigenStepper:

```
// Set up EigenStepper with  given B-Field 'm_cfg.bField'
Acts::EigenStepper<> stepper(m_cfg.bField);
```

```
// Set up the propagator
using Propagator = Acts::Propagator<Acts::EigenStepper<>>;
auto propagator = std::make_shared<Propagator>(stepper);
```

Setting up required tools for the vertex fitter

Now, set up an impact point estimator...

```
// Set up ImpactPointEstimator
using IPEstimator = Acts::ImpactPointEstimator<Acts::BoundTrackParameters,
    Propagator>;
IPEstimator::Config ipEstimatorCfg(m_cfg.bField, propagator);
IPEstimator ipEstimator(ipEstimatorCfg);
```

... and track linearizer for helical track parameters:

```
// Set up the helical track linearizer
using Linearizer = Acts::HelicalTrackLinearizer<Propagator>;
Linearizer::Config ltConfig(m_cfg.bField, propagator);
Linearizer linearizer(ltConfig);
```

Now, for the sake of this example, let's specify a user-defined annealing scheme for the AVMF:

```
// Set up deterministic annealing with user-defined temperatures
std::vector<double> temperatures{8.0, 4.0, 2.0, 1.4142136, 1.2247449,
    1.0};
Acts::AnnealingUtility::Config annealingConfig(temperatures);
Acts::AnnealingUtility annealingUtility(annealingConfig);
```

The AMVF strongly interplays with its dedicated vertex fitter, the Adaptive Multi-Vertex Fitter. Let's configure and set it up with the annealing utility defined above:

```
// Set up the vertex fitter with user-defined annealing
using Fitter = Acts::AdaptiveMultiVertexFitter<
  Acts::BoundTrackParameters, Linearizer>;
Fitter::Config fitterCfg(ipEstimator);
fitterCfg.annealingTool = annealingUtility;
Fitter fitter(fitterCfg);
```

Setting up required tools: vertex seed finder

The last tool we need to set up (before finally setting up the AMVF) is a vertex seed finder:

```
// Set up the vertex seed finder
using SeedFinder = Acts::TrackDensityVertexFinder<Fitter, Acts::
    GaussianTrackDensity<Acts::BoundTrackParameters>>;
SeedFinder seedFinder;
```

Setting up the AMVF tool

Now we are ready to set up the Adaptive Multi-Vertex Finder. ACTS vertex finders are templated on the vertex fitter and vertex seed finder type:

```
// The vertex finder type
using Finder = Acts::AdaptiveMultiVertexFinder<Fitter, SeedFinder>;
```

We configure the vertex finder in such a way that we do not use a beam spot constraint here:

```
// The finder configuration
Finder::Config finderConfig(std::move(fitter), seedFinder, ipEstimator,
    linearizer);
// We do not want to use a beamspot constraint here
finderConfig.useBeamSpotConstraint = false;
```

Create the AMVF instance and a finder state to be passed to the `find()` method below:

```
// Instantiate the finder
Finder finder(finderConfig);
// The vertex finder state
Finder::State state;
```

Lastly, we need to provide vertexing options. Here, we could e.g. set a beam spot constraint to the vertexing

```
// Default vertexing options
using VertexingOptions = Acts::VertexingOptions<Acts::BoundTrackParameters
    >;
VertexingOptions finderOpts(ctx.geoContext, ctx.magFieldContext);
```

Deploying the vertex finder on the track collection

Now we're ready to actually use the AMVF tool that we have set up above to find vertices on our input track collection. The `find()` methods on ACTS vertex finders return an `Acts::Result` object that we can use to check if any errors occurred and to retrieve the vertex collection:

```
// Find vertices
auto res = finder.find(inputTrackPointers, finderOpts, state);

if (res.ok()) {
  // Retrieve vertices found by vertex finder
  auto vertexCollection = *res;
  ACTS_INFO("Found " << vertexCollection.size() << " vertices in event.");

  unsigned int count = 0;
  for (const auto& vtx : vertexCollection) {
    ACTS_INFO("\t" << ++count << ". vertex at "
                   << "(" << vtx.position().x()
                   << "," << vtx.position().y()
                   << "," << vtx.position().z()
```

```
                    << ") with " << vtx.tracks().size() << " tracks.");
  }
} else {
  ACTS_ERROR("Error in vertex finder: " << res.error().message());
}
```

A.5 AMVF and IVF Configuration for ACTS Standalone Studies

Configuration parameter	Value
useBeamSpotConstraint	false (true)
tracksMaxZinterval	3 mm
tracksMaxSignificance	5
maxVertexChi2	18.42
doRealMultiVertex	true
maxMergeVertexSignificance	3
minWeight	0.0001
maxIterations	200
useSeedConstraint	false

Table A.1: AMVF configuration parameters for ACTS standalone studies. Unless stated otherwise, the listed parameters correspond to the ATLAS Run 3 configuration parameters. If parameters deviating from this configuration are used, the ATLAS default configuration is given in brackets.

Configuration parameter	Value
useBeamSpotConstraint	false (true)
significanceCutSeeding	10
maximumChi2cutForSeeding	36
maxVertices	200 (50)
createSplitVertices	false
doMaxTracksCut	false
maxTracks	5000
cutOffTrackWeight	0.01

Table A.2: IVF configuration parameters for ACTS standalone studies. Unless stated otherwise, the listed parameters correspond to the ATLAS Run 2 configuration parameters. If parameters deviating from this configuration are used, the ATLAS default configuration is given in brackets.

A.6 Determination and Overview of (Adaptive) Grid Seeder Configuration Parameters

The size and shape of the track density grid representation in the (Adaptive) Grid Seeder largely depend on the track's impact parameter uncertainties. The larger the uncertainties, the broader the density grid representation. In order to determine an adequate track grid size, the impact parameter uncertainty distributions of all input tracks to the primary vertexing in 3000 $t\bar{t}$ $\langle\mu\rangle = 40 - 60$ events are shown in Fig.A.4, Fig. A.5 and Fig. A.6.
As can be seen by comparing Fig.A.4 and Fig. A.5, the z_0 uncertainty exhibits much larger values than its d_0 counterpart, resulting in the z_0 uncertainties being the dominant contributor to the track density grid shape. The chosen track grid sizes must be large enough to accommodate the majority of occurring uncertainties and therefore track density shapes but should still be small enough to guarantee a good CPU performance of the (Adaptive) Grid Seeder algorithm. It was therefore decided to choose the configuration parameters in such way that the majority of track density values of all tracks with $\sigma^2(z_0, z_0) < 2\,\text{mm}^2$ will be fully contained within the density grid representation. If track density grids (or rather the overlap vectors with the beam axis) need to be generated in the rare case of tracks with $\sigma^2(z_0, z_0) > 2\,\text{mm}^2$, the outermost tails of the track density distribution are not fully contained in the density grid anymore. As these cases do not occur very often and only disregard minor density contributions from the tails of the distributions, the performance impact should be negligible.

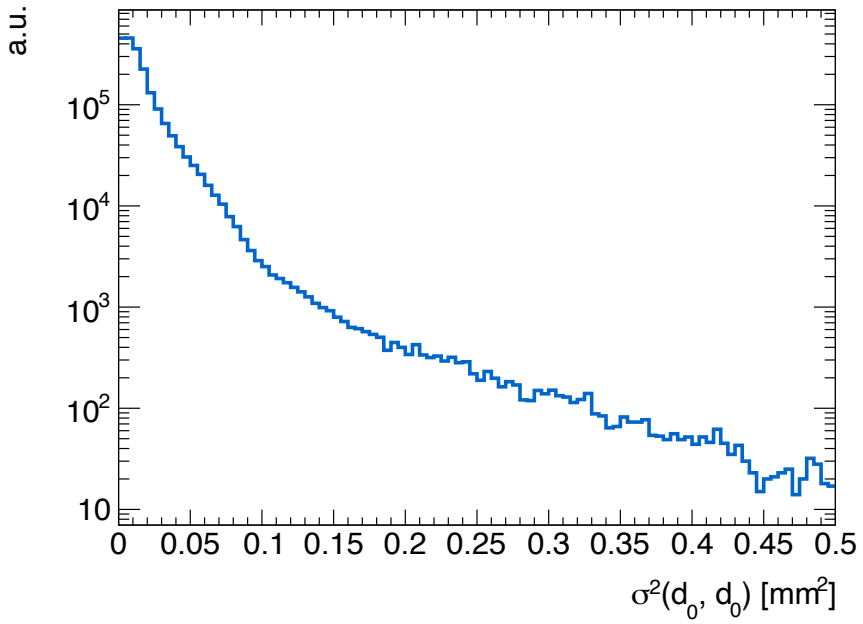


Figure A.4: Distribution of the $\sigma^2(d_0, d_0)$ impact parameter uncertainty of all input tracks to the primary vertex reconstruction in $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

A track density representation with $\sigma^2(z_0, z_0) = 2\,\text{mm}^2$ is shown in Fig. A.7. In order to

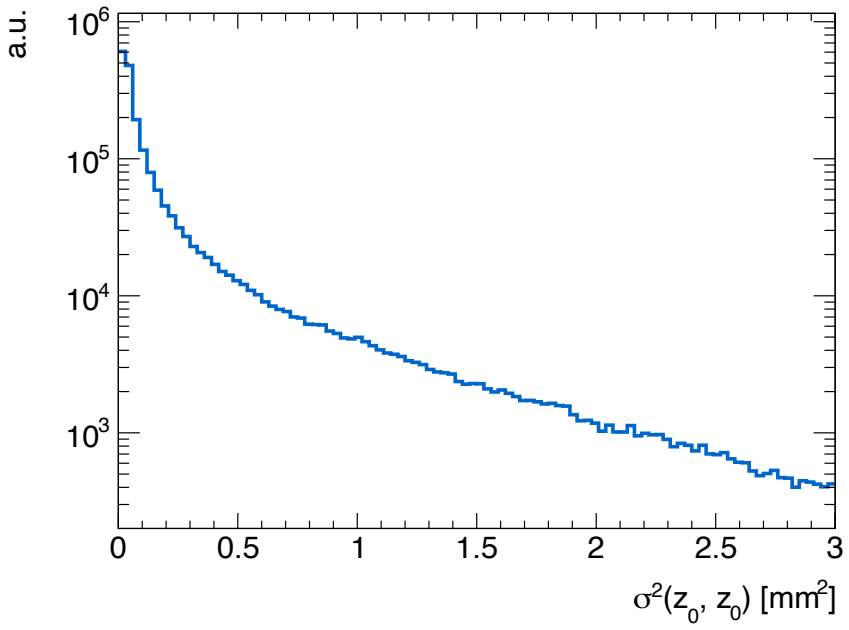


Figure A.5: Distribution of the $\sigma^2(z_0, z_0)$ impact parameter uncertainty of all input tracks to the primary vertex reconstruction in $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

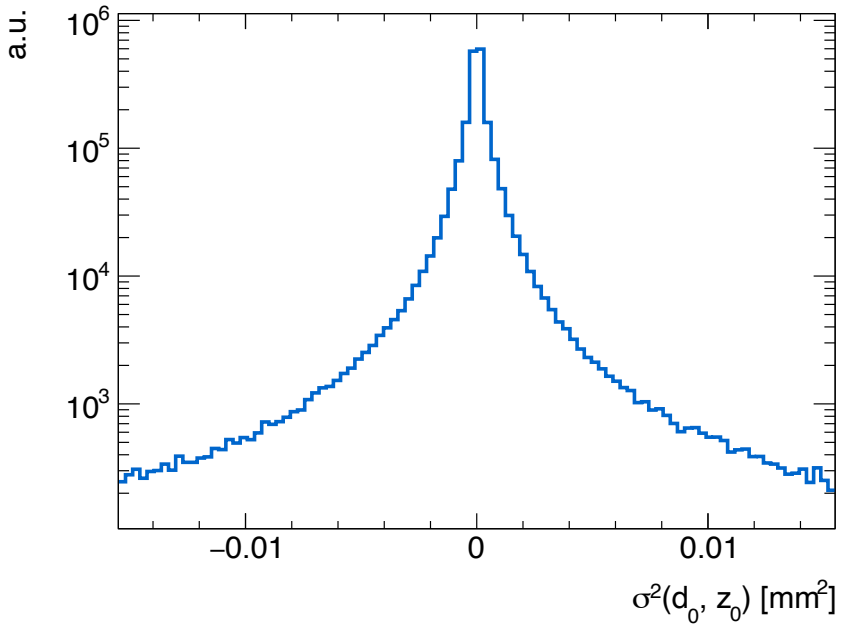


Figure A.6: Distribution of the $\sigma^2(d_0, z_0)$ impact parameter uncertainty of all input tracks to the primary vertex reconstruction in $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.

fully contain the vast majority of the density distribution, a grid size of $N_{\text{bins}}^{\text{trk}} \times N_{\text{bins}}^{\text{trk}} = 55 \times 55$ bins[1] with a bin size of $w = 0.1\,\text{mm}$ is needed. The full density grid therefore exhibits a spatial extent of $5.5\,\text{mm} \times 5.5\,\text{mm}$, an approximate value that always should be aimed for when selecting w and $N_{\text{bins}}^{\text{trk}}$.

[1] Note that for simplicity only squared track grids are used.

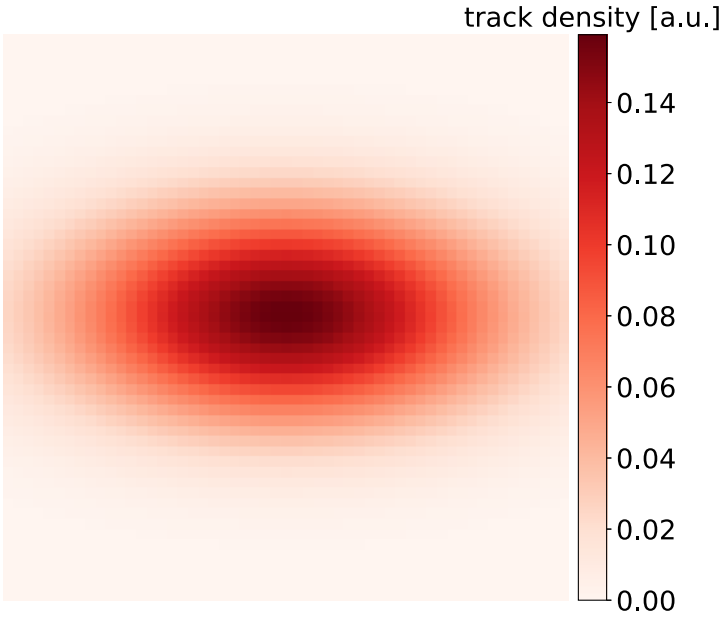
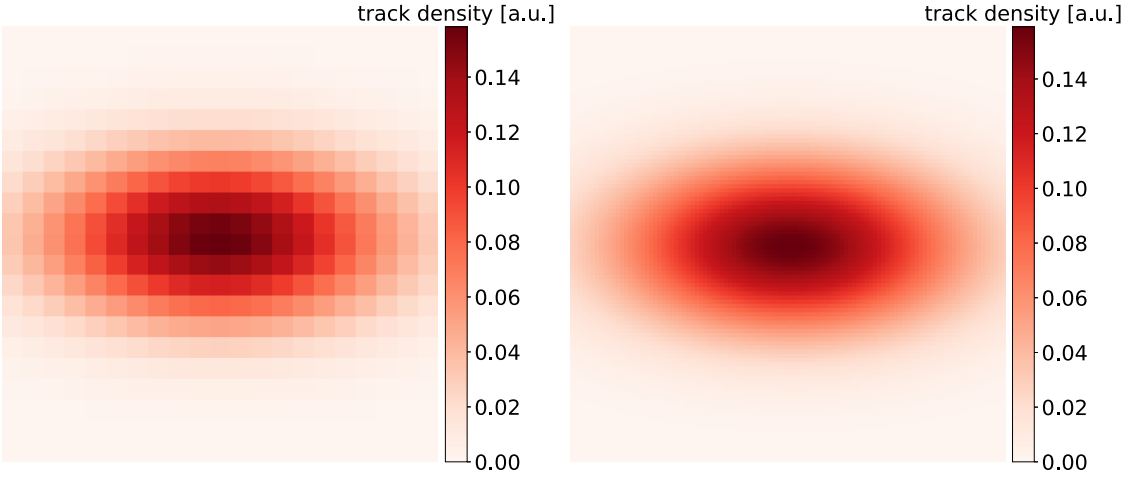


Figure A.7: Example grid density representation for a track with $\sigma^2(d_0, d_0) = 0.5\,\text{mm}^2$, $\sigma^2(z_0, z_0) = 2\,\text{mm}^2$ and $\sigma^2(d_0, z_0) = -0.01\,\text{mm}^2$. The grid consists of 55×55 bins of size 0.1 mm.

Fig. A.8 shows two more track density representations on a grid for the same track, using different w and $N_{\text{bins}}^{\text{trk}}$ configurations, all aiming for a coverage greater than 5 mm × 5 mm and therefore largely containing the full track density distribution.
Table A.3 gives an overview over all different (Adaptive) Grid Seeder configurations that are used throughout this thesis and in particular in Chapter 7 when its physics and CPU performance is evaluated.



(a) (b)

Figure A.8: Examples of two grid density representations of the same track with a grid size of (a) $N_{\text{bins}}^{\text{trk}} \times N_{\text{bins}}^{\text{trk}} = 21 \times 21$ bins and bin width of $w = 0.25\,\text{mm}$ and (b) $N_{\text{bins}}^{\text{trk}} \times N_{\text{bins}}^{\text{trk}} = 109 \times 109$ bins and bin width of $w = 0.05\,\text{mm}$.

w	$N_{\text{bins}}^{\text{trk}}$	z_{max}	N_{bins}
0.025 mm	219	250 mm	20 000
0.050 mm	109	250 mm	10 000
0.100 mm	55	250 mm	5000
0.250 mm	21	250 mm	2000
0.500 mm	11	250 mm	1000

Table A.3: Overview of the (Adaptive) Grid Seeder configuration parameters used throughout Chapter 7, unless stated otherwise. The z_{max} and N_{bins} values only refer to the non-adaptive version where a maximum z-coverage along the beam axis and the corresponding number of bins are required.

A.7 ACTS Vertexing Integration in the ATLAS Software Framework

The integration of the ACTS vertexing into the ATLAS software framework required the implementations of `const`-correct and hence thread-safe Athena wrapping tools for both the ACTS AMVF and ACTS IVF, provided as the `ActsAdaptiveMultiPriVtxFinderTool` as well as the `ActsIterativePriVtxFinderTool` which can be found in the Athena repository in `Tracking/Acts/ActsPriVtxFinder`. The general structure of these tools can be subdivided into four stages:

1. **ATLAS-to-ACTS track EDM conversion:**
   First, the ATLAS-given track objects of type `Trk::ITrackLink` need to be converted into an ACTS-readable format. As explained in Appendix A.1.2, any user-defined track type can be used as input to the ACTS vertexing, provided that `Acts::BoundTrackParameters` objects can be retrieved from it. Therefore, a dedicated track wrapper class was defined in Athena that allows to retain the original ATLAS track link information throughout and after the ACTS primary vertex reconstruction process. Listing A.4 shows the implementation details of the track wrapper class and therefore an example application of how the ACTS vertexing EDM allows for flexible user inputs in an experiment-independent fashion.

2. **ACTS vertex finder call:**
   After the set of input tracks has been converted into `TrackWrapper` types and the ACTS vertexing was configured and set up, the ACTS vertex finder is invoked to reconstruct the primary vertices.

3. **ACTS-to-ATLAS vertex EDM conversion:**
   The output list of reconstructed ACTS vertices needs to be reconverted into ATLAS vertex objects of type `xAOD::Vertex`, associating the retained `Trk::ITrackLink` (ATLAS) objects to their corresponding reconstructed vertex.

4. **Hard-scatter selection:**

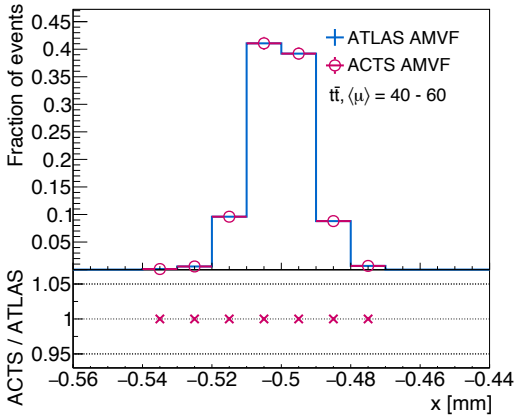
The list of all reconstructed primary vertices is sorted based on their compatibility with being the HS vertex (i.e. by default the vertex with the highest $\sum p_T^2$ of all associated tracks). The first element in the sorted list is hence considered the selected HS vertex.

A seamless integration of the ACTS primary vertex finder tools into the existing ATLAS primary vertex reconstruction framework was achieved by making the ACTS wrapper tools inherit from the `InDet::IVertexFinder` interface, which has been made `const`-correct in order to ensure a thread-safe reconstruction process.
For the same reason, the `InnerDetector/InDetRecAlgs/InDetPriVxFinder` algorithm was rendered `const`-correct and now inherits from `AthReentrantAlgorithm` to allow a multi-threaded execution.
An `InDetFlag` called `useActsPriVertexing` was introduced, which enables (when set to `true`) the utilization of the ACTS primary vertexing. The entire corresponding tool configuration and setup has been implemented in the `InDetRecExample`.

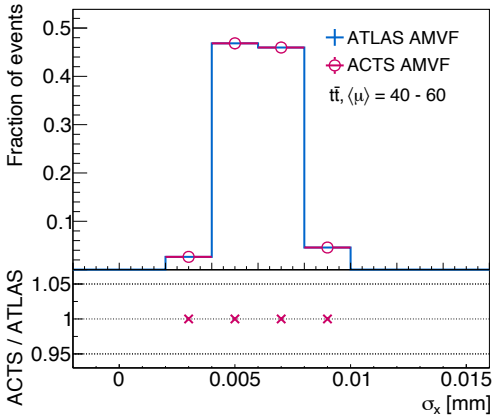
```
class TrackWrapper {
public:
  /// Constructor
  TrackWrapper(const Trk::ITrackLink* trkLink,
               const Acts::BoundTrackParameters& boundParams)
    : m_trkLink(trkLink)
    , m_boundParams(boundParams)
    {}

  /// BoundTrackParameters getter
  const Acts::BoundTrackParameters& parameters() const
  {
    return m_boundParams;
  }

  /// Track link getter
  const Trk::ITrackLink* trackLink() const
  {
    return m_trkLink;
  }

private:
  /// Original track link object in ATLAS
  const Trk::ITrackLink* m_trkLink;
  /// Track parameter representation in ACTS
  Acts::BoundTrackParameters m_boundParams;
};
```

Listing A.4: Definition of the track wrapper type used as input to the ACTS primary vertexing tools in Athena.

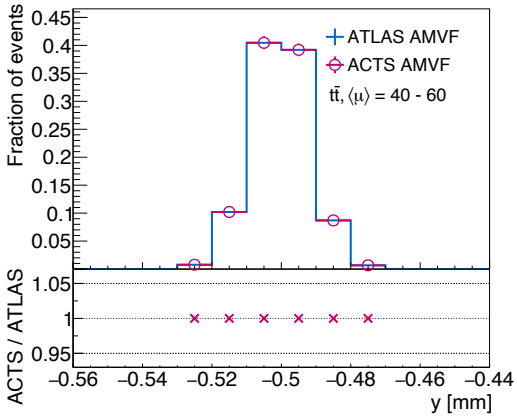
A.8 ACTS Vertexing in ATLAS – Single-threaded AMVF Validation



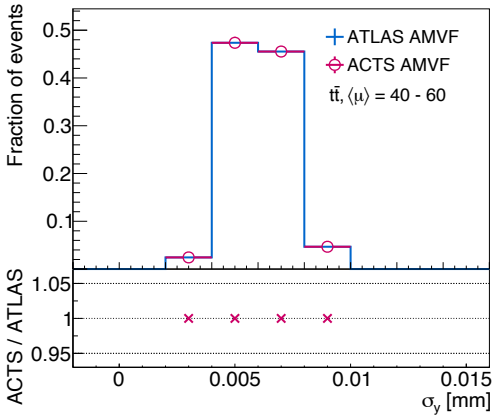
(a) Selected HS vertex – x-positions



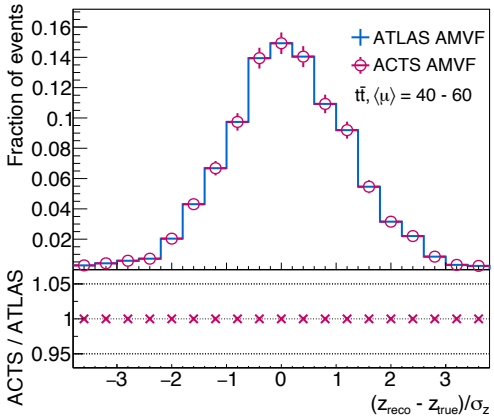
(b) Selected HS vertex – x-position errors



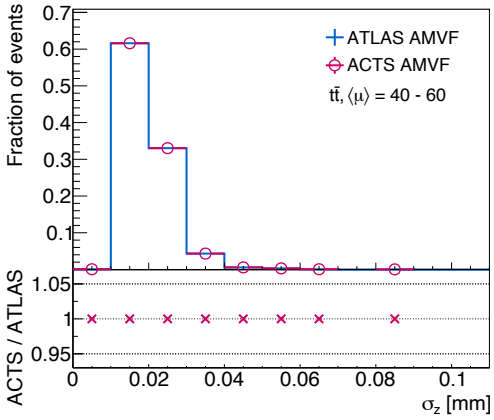
(c) Selected HS vertex – y-positions



(d) Selected HS vertex – y-position errors



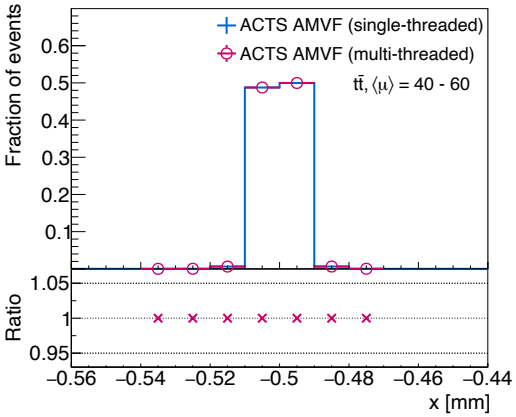
(e) Selected HS vertex – z-pull



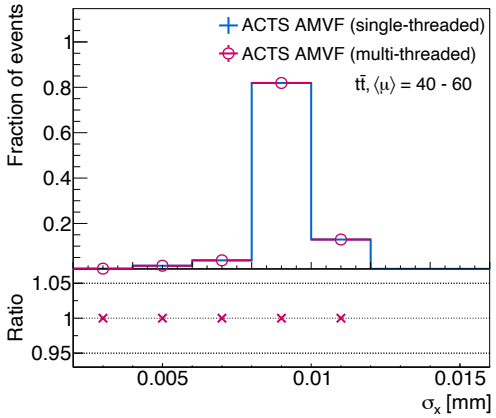
(f) Selected HS vertex – z-position errors

Figure A.9: Comparison of the ACTS AMVF and ATLAS AMVF implementations, showing various different variables for the selected HS vertices in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events in single-threaded execution mode. The HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.

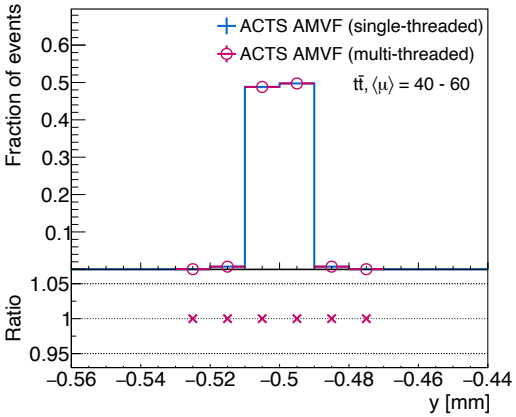
A.9 ACTS Vertexing in ATLAS – Multi-threaded AMVF Validation



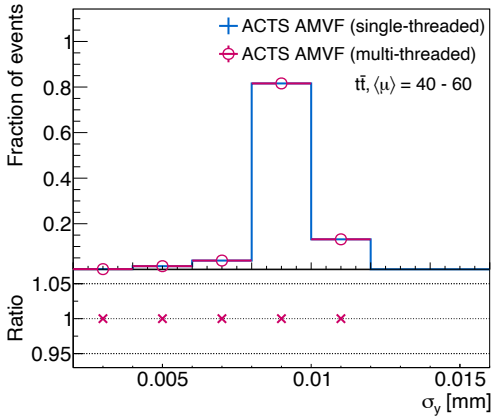
(a) All vertices – x-positions



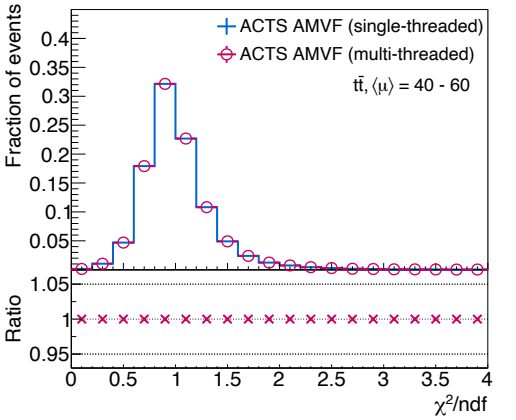
(b) All vertices – x-position errors



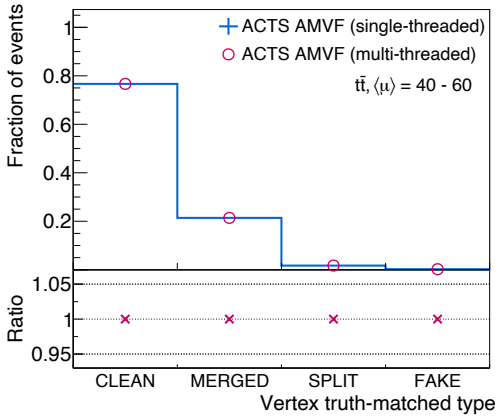
(c) All vertices – y-positions



(d) All vertices – y-position errors

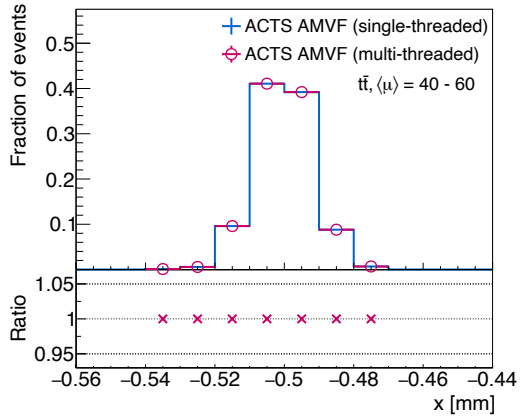


(e) All vertices – reduced χ^2

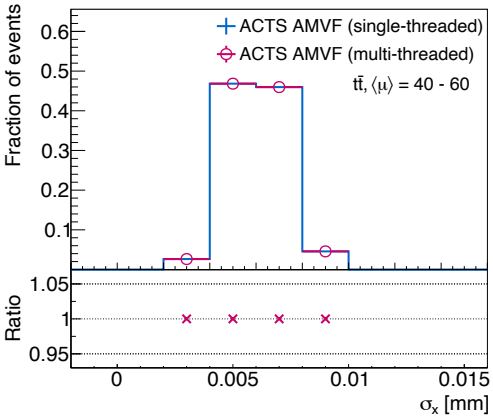


(f) All vertices – truth-matched type

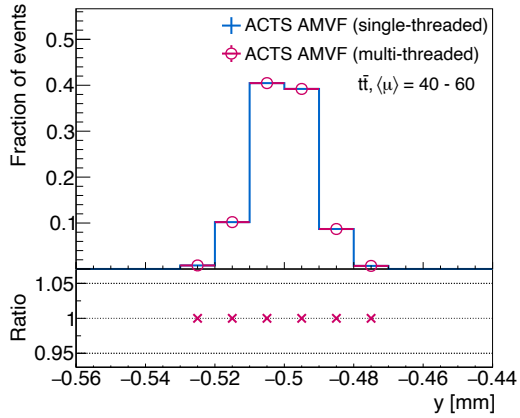
Figure A.10: Comparison of the ACTS AMVF algorithm in single-threaded and multi-threaded execution mode, showing various different variables for all primary vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$. The multi-threaded execution was performed on eight concurrent threads.



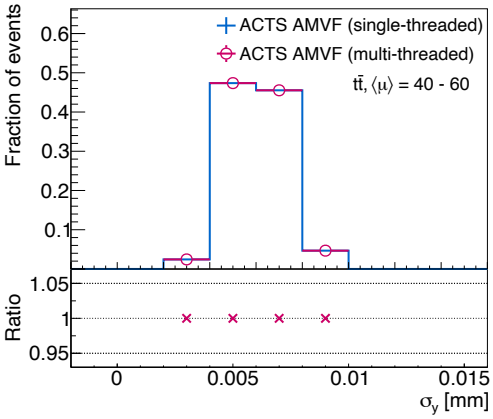
(a) Selected HS vertex – x-positions



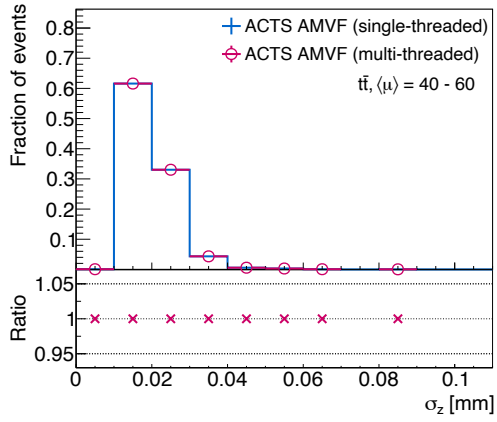
(b) Selected HS vertex – x-position errors



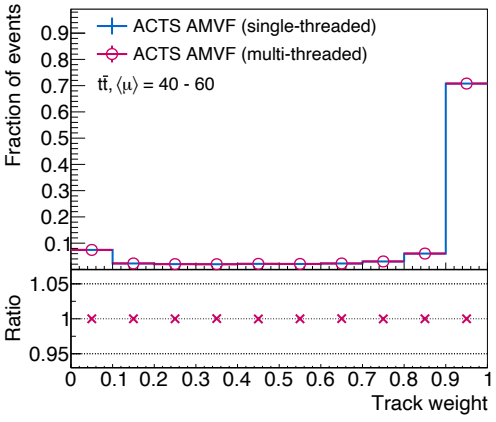
(c) Selected HS vertex – y-positions



(d) Selected HS vertex – y-position errors

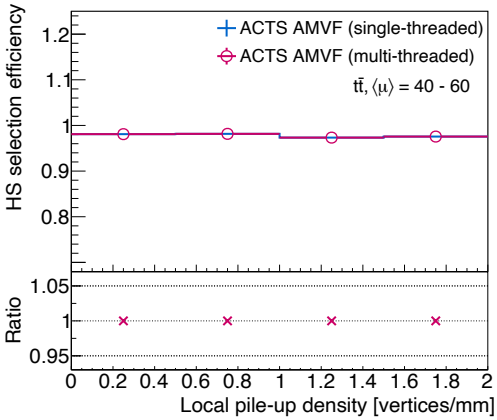


(e) Selected HS vertex – z-position errors

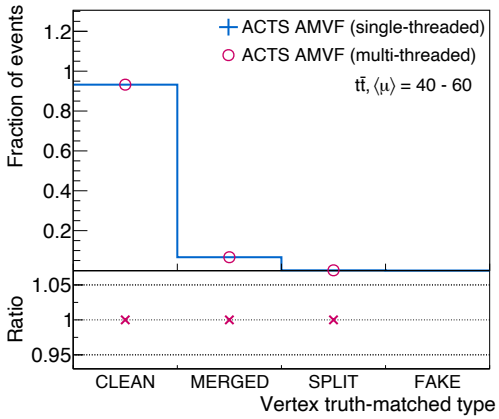


(f) Selected HS vertex – track weights

Figure A.11: Comparison of the ACTS AMVF algorithm in single-threaded and multi-threaded execution mode, showing various different variables for the selected HS vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$. The multi-threaded execution was performed on eight concurrent threads and the HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.



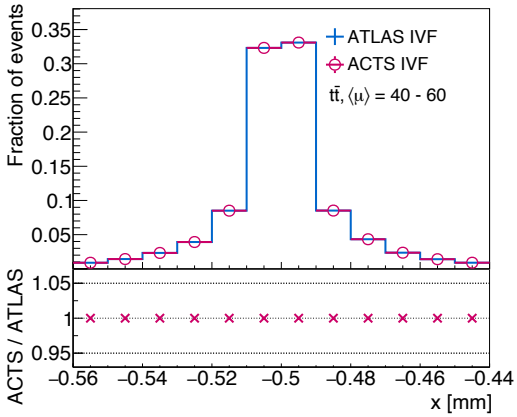
(a) HS selection efficiency



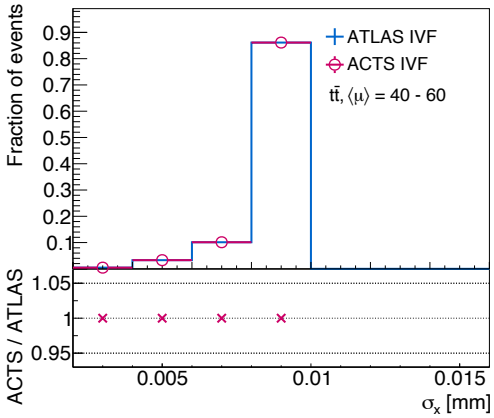
(b) Selected HS vertex – truth-matched type

Figure A.12: Comparison of the ACTS AMVF algorithm in single-threaded and multi-threaded execution mode, showing various different variables for the selected HS vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $\mathrm{t\bar{t}}$. The HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.

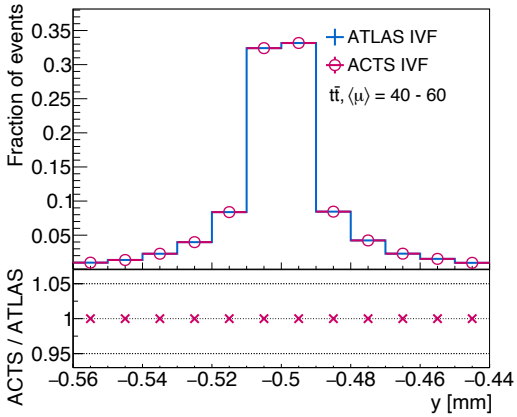
A.10 ACTS Vertexing in ATLAS – Single-threaded IVF Validation



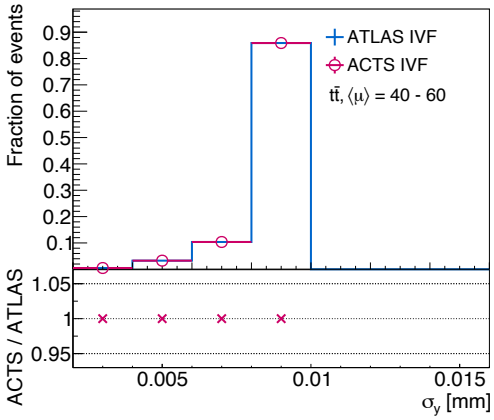
(a) All vertices – x-positions



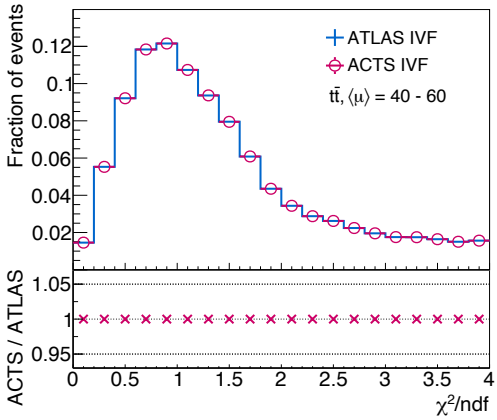
(b) All vertices – x-position errors



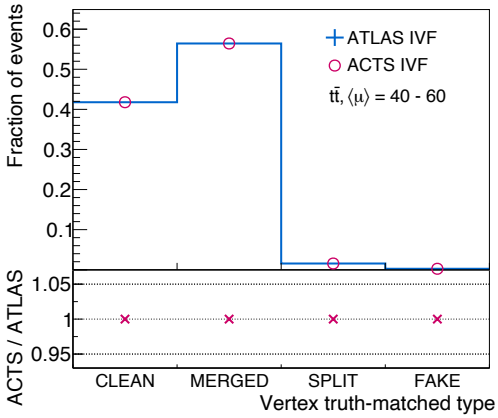
(c) All vertices – y-positions



(d) All vertices – y-position errors

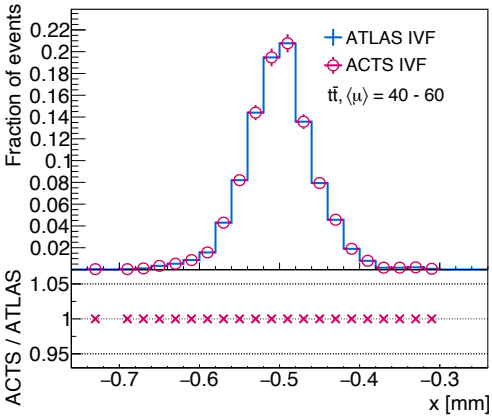


(e) All vertices – reduced χ^2

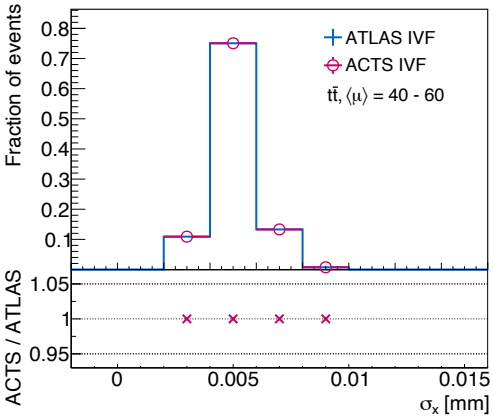


(f) All vertices – truth-matched type

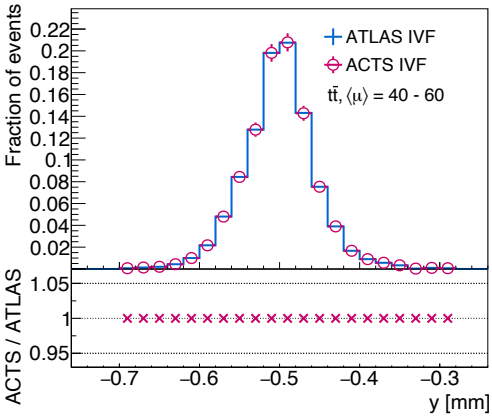
Figure A.13: Comparison of the ACTS IVF and ATLAS IVF implementations, showing various different variables for all primary vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events in single-threaded execution mode.



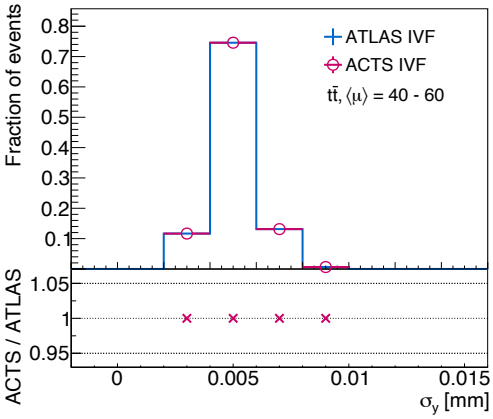
(a) Selected HS vertex – x-positions



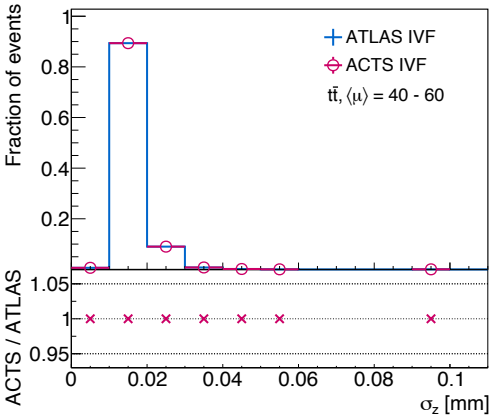
(b) Selected HS vertex – x-position errors



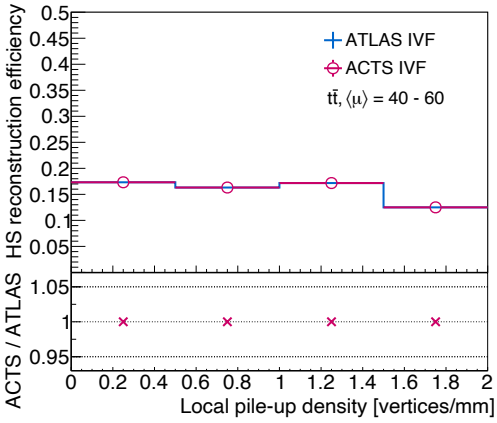
(c) Selected HS vertex – y-positions



(d) Selected HS vertex – y-position errors



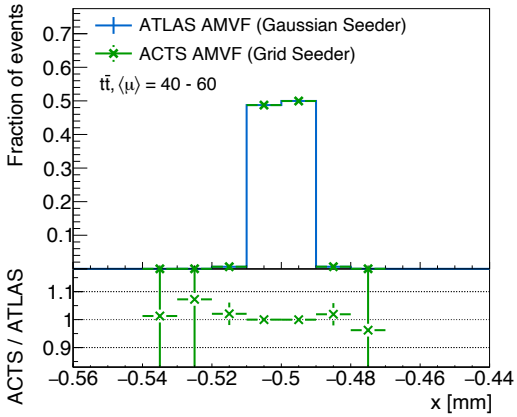
(e) Selected HS vertex – z-position errors



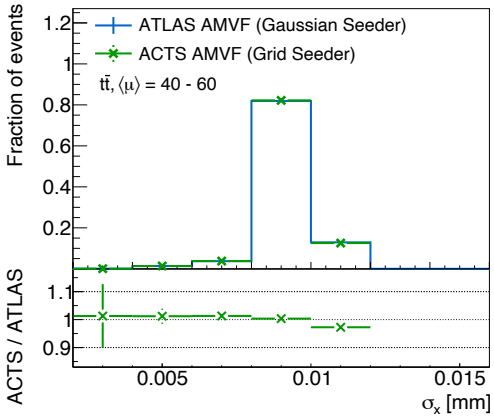
(f) HS reconstruction efficiency

Figure A.14: Comparison of the ACTS IVF and ATLAS IVF implementations, showing various different variables for the selected HS vertices in 3000 $\langle\mu\rangle = 40-60$ $\mathrm{t\bar{t}}$ $t\bar{t}$ events in single-threaded execution mode. The HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.

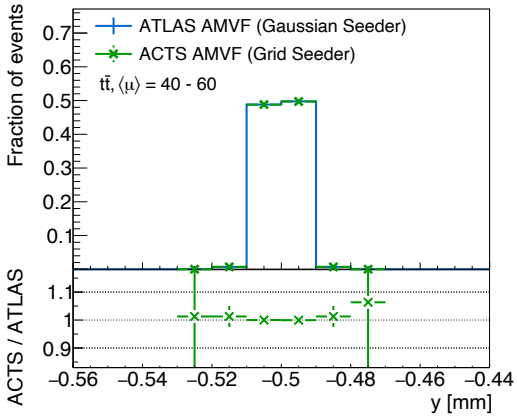
A.11 ACTS Vertexing in ATLAS – AMVF Grid Seeder Performance



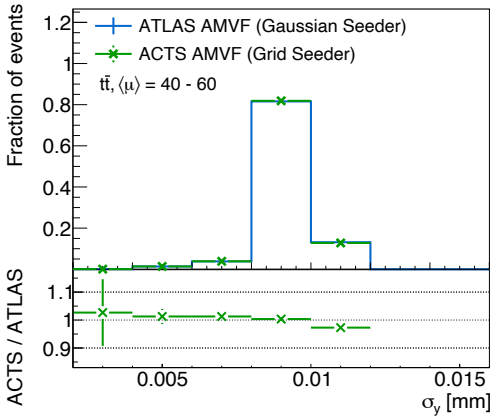
(a) All vertices – x-positions



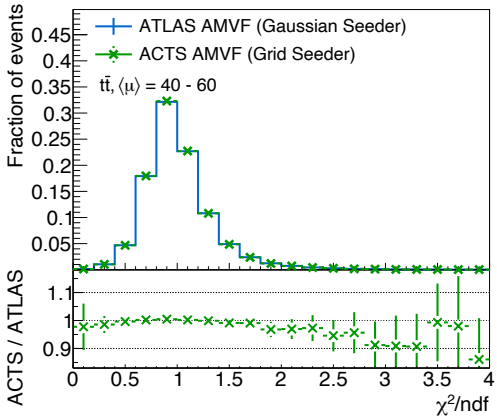
(b) All vertices – x-position errors



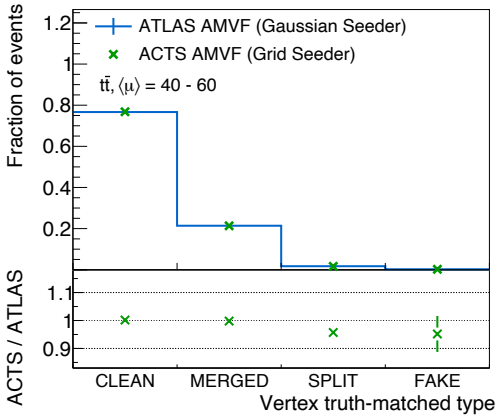
(c) All vertices – y-positions



(d) All vertices – y-position errors

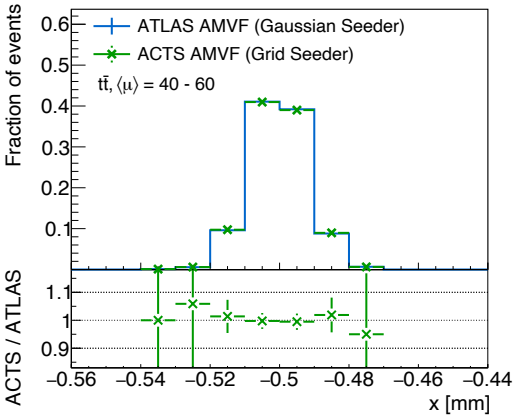


(e) All vertices – reduced χ^2

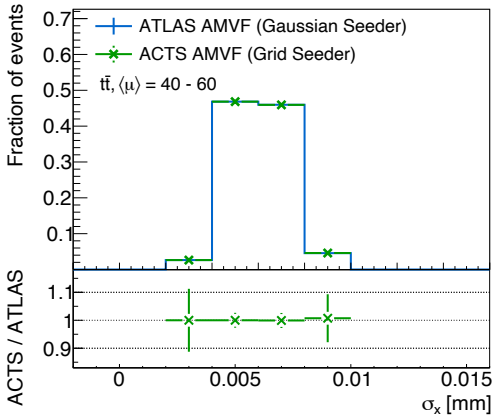


(f) All vertices – truth-matched type

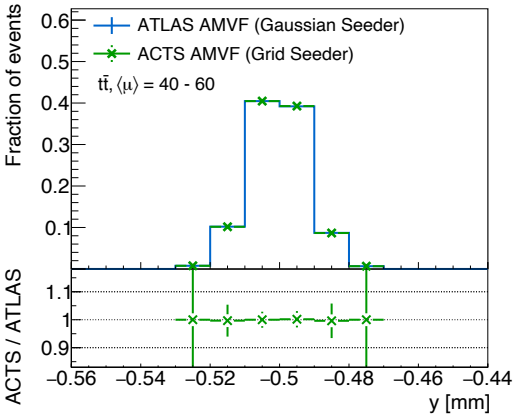
Figure A.15: Comparison of the ATLAS AMVF algorithm utilizing the Gaussian Seeder and the ACTS AMVF algorithm utilizing a $w = 0.05\,\text{mm}$ Grid Seeder configuration, showing various different variables for all primary vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $\text{t}\bar{\text{t}}$ events in single-threaded execution mode.



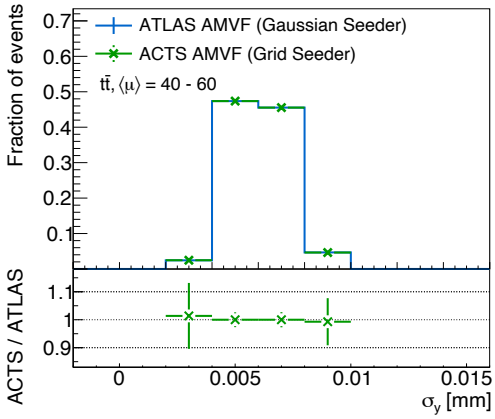
(a) Selected HS vertex – x-positions



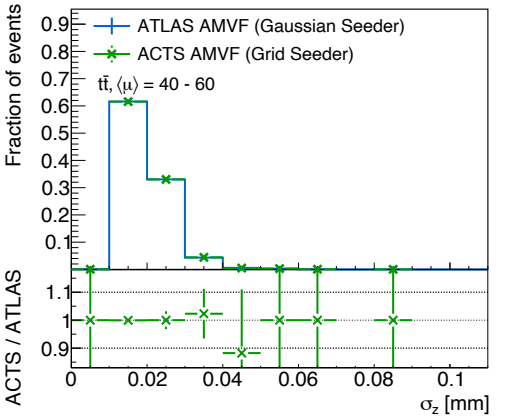
(b) Selected HS vertex – x-position errors



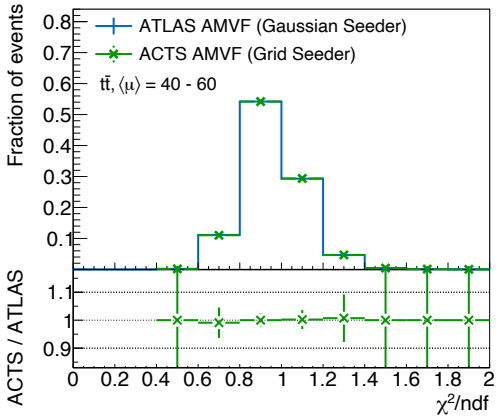
(c) Selected HS vertex – y-positions



(d) Selected HS vertex – y-position errors



(e) Selected HS vertex – z-position errors



(f) Selected HS vertex – reduced χ^2

Figure A.16: Comparison of the ATLAS AMVF algorithm utilizing the Gaussian Seeder and the ACTS AMVF algorithm utilizing a $w = 0.05\,\text{mm}$ Grid Seeder configuration, showing various different variables for the selected HS vertices reconstructed in 3000 $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events in single-threaded execution mode. The HS vertex is selected as the vertex with the highest $\sum p_T^2$ of all associated tracks.

A.12 Classification of Merged Vertices using Deep Neural Networks

A.12.1 Introduction

In high pile-up environments, primary interaction vertices can be located very close to one another and can therefore mistakenly be reconstructed as one single, merged vertex. In Chapter 5, Fig. 5.4 showed the average number of reconstructed primary vertices by the AMVF as a function of the number of true pp interactions per bunch-crossing in $t\bar{t}$ events together with their classifications in the classes MATCHED, MERGED, SPLIT and FAKE. For large numbers of simultaneous pp interactions, more than 20% of all reconstructed vertices show significant track contributions from more than one simulated interaction and are thus classified as MERGED vertices.
While the merging of reconstructed primary vertices is not necessarily a problem for physics analyses if it happens purely among pile-up vertices, it can have a significant physics performance impact in case the hard-scatter vertex undergoes merging with nearby pile-up vertices (pile-up contamination). A classification method to detect merged (hard-scatter) vertices, employed immediately after the full AMVF reconstruction step, would therefore be very beneficial to mitigate these undesirable effects. The information of a primary (hard-scatter) vertex being classified as MERGED during reconstruction could then be used to initiate a dedicated resolving procedure which tries to separately reconstruct these merged vertices or, in the most simple case, provides an analysis with information based on which they can decide to discard the event if the hard-scatter vertex has pile-up contamination.
The following briefly discusses the development and deployment of a neural network architecture, developed in the course of this thesis, used to classify AMVF-reconstructed primary vertices with the goal to reliably detect pile-up contamination in reconstructed primary hard-scatter vertices.

A.12.2 Neural Network Architecture

In order to understand if a vertex is a merged vertex, i.e. understand if it contains a significant number of tracks from neighboring pp interactions, its track content needs to be evaluated. Thus, relevant track parameter information of each track associated to the vertex of interest needs to be fed to the neural network as input. Each input track to the neural network is chosen to be presented by a seven-dimensional feature vector

$$v = (d_0, |z_{\text{vertex}} - z_0|, \eta, p_{\text{T}}, \sigma(d_0), \sigma(z_0), \chi^2/\text{ndf}) \tag{A.4}$$

with the track's transverse and longitudinal impact parameters d_0 and z_0, respectively, their associated uncertainties $\sigma(d_0)$ and $\sigma(z_0)$, the reconstructed vertex z-position z_{vertex}, the track's pseudo-rapidity η and transverse momentum p_{T} as well as its reduced χ^2 value, indicating the track fit quality.
Since the grouping of tracks to vertices in vertex reconstruction can be understood as a

pattern recognition task, two different types of neural network architectures that have originally been designed for pattern recognition, recurrent neural networks (RNNs) [184] as well as (one-dimensional) convolutional neural networks (CNNs) [185], have been tested. An RNN architecture comes with the advantage that it is very well suited for dynamic input lengths, as needed for the variable number of tracks associated to a vertex in merged vertex classification. Its training and execution, however, usually takes significantly longer than for a CNN model. The static architecture of a CNN, on the other hand, demands a fixed input size, i.e. a fixed and thus maximum number of input tracks, which can be realized if a large enough input zero-padding is chosen.

As both developed architecture were found to be almost equally well suited for solving the given problem with the CNN model always slightly outperforming the RNN in terms of accuracy and efficiency, the following will exclusively focus on the developed CNN model and its performance.

Fig. A.17 schematically shows the deployed one-dimensional CNN architecture. The input is a grid of size 7×200 (seven track features and a chosen maximum number of 200 considered tracks per vertex), while the desired neural network output score is a single number in the range of $[0, 1]$, 0 indicating a clean, unmerged vertex and 1 a merged vertex. The hidden layers comprise two convolutional layers with 400 filters each, which alternate with two average pooling layers, followed by a final flatten and three densely connected layers with 300 neurons each. ReLU activation functions as well as a 20% dropout regularization is used in every layer.

The network is trained on 1.2 million AMVF-reconstructed vertices with truth vertex class information from $\langle\mu\rangle = 40 - 60$ $\mathrm{t\bar{t}}$ events using a binary crossentropy loss function.

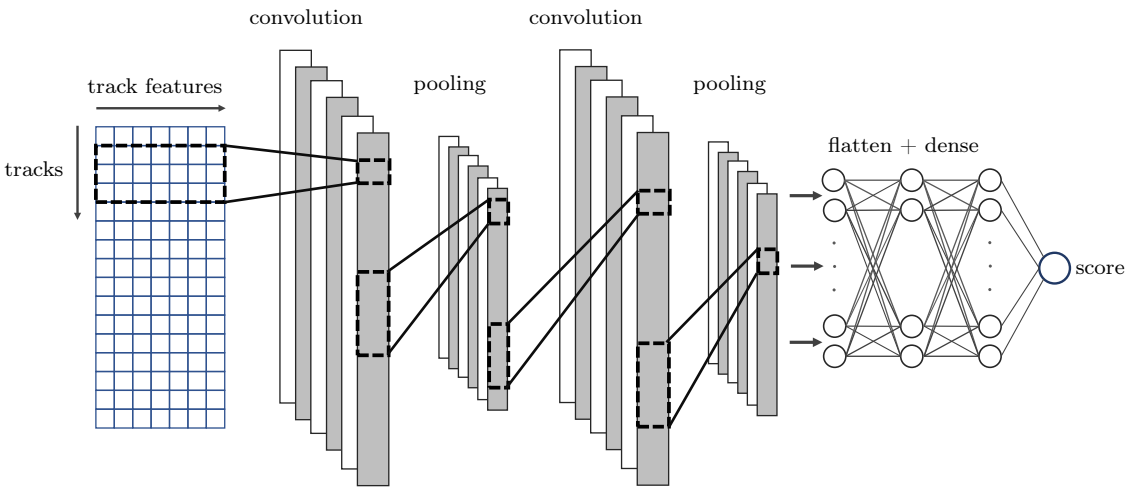
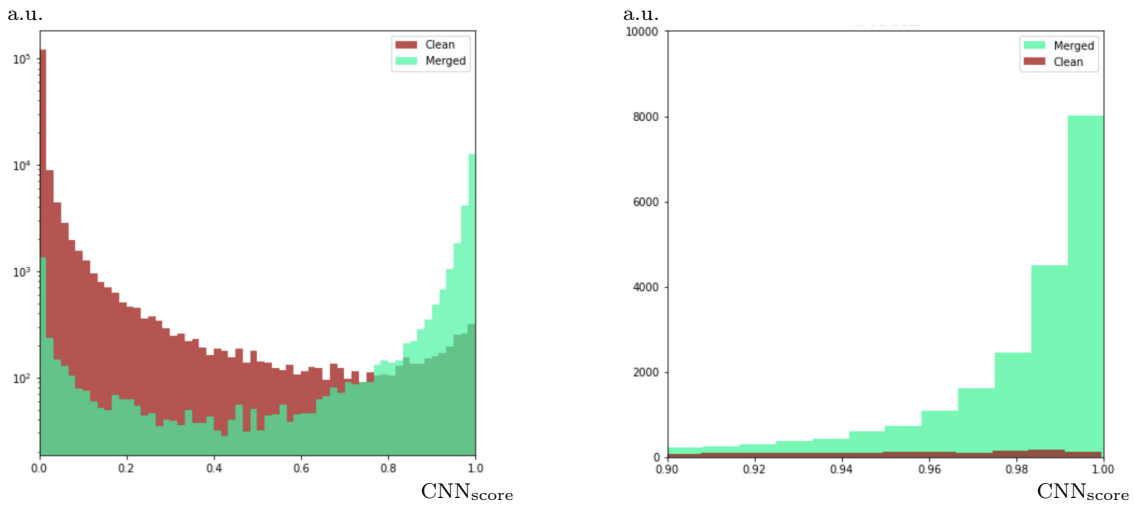


Figure A.17: Schematic illustration of the developed one-dimensional CNN model architecture for the classification of merged vertices. The one-dimensional CNN applied on two-dimensional input data only allows to slide the NN kernel (of size 7×3 in the chosen architecture) in one single direction, forcing it to analyze all seven track features of a series of tracks simultaneously.

A.12.3 Results

The network's performance is evaluated on a test dataset, comprising 2×10^5 previously unseen AMVF-reconstructed vertices from $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events.
Fig. A.18 shows the output score distribution for all 2×10^5 AMVF-reconstructed primary vertices as well as their true vertex classes in red (clean) and green (merged). A very good classification and separation power can be seen with the classifiers true positive rate[2] (TPR) of TPR = 0.88 and true negative rate[3] (TNR) of TNR = 0.97.
An extremely pure environment is observed in Fig. A.19 for the important case of classifying the reconstructed hard-scatter vertex. The CNN is able to separate the different vertex classes very efficiently and detects merged hard-scatter vertices with very high accuracy. In this case, a TPR = 0.98 and TNR = 0.99 is observed.
Due to the lack of suitable MC samples, further studies (e.g. on different processes and in different pile-up conditions) have not been conducted in the scope of this thesis. However, as the presented method shows very promising results in $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events, follow-up studies should be conducted to investigate a potential deployment in ATLAS primary vertex reconstruction. Based on this CNN classification method, physics analyses could then decide to discard events with merged hard-scatter vertices or try to resolve the vertices in order to further increase the physics performance.

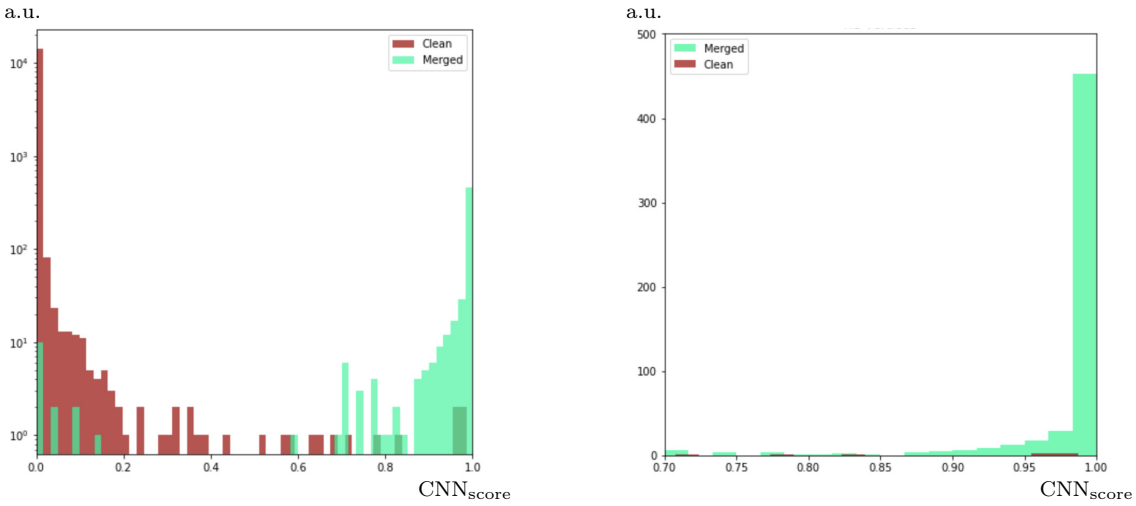


(a) (b)

Figure A.18: (a) CNN merged-vertex classifier output score distribution for all 2×10^5 AMVF-reconstructed primary vertices in $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events, showing the true vertex classes in red (clean) and green (merged). (b) Zoomed-in view showing the classification results for an output score $CNN_{score} > 0.9$. Note that a logarithmic scale is chosen for the y-axis in (a) while (b) uses a linear scale.

[2]The true positive rate is defined as the ratio of true positive classifications and all possible true classifications and can be seen as the probability that the CNN classifies the vertex as MERGED given the vertex is indeed a merged vertex.

[3]The true negative rate is defined as the ratio of true negative classifications and all possible negative classifications and can be seen as the probability that the CNN classifies the vertex as CLEAN given the vertex is indeed a clean vertex.



(a) (b)

Figure A.19: (a) CNN merged-vertex classifier output score distribution for all hard-scatter vertices among the 2×10^5 AMVF-reconstructed primary vertices in $\langle\mu\rangle = 40 - 60$ $t\bar{t}$ events, showing the true vertex classes in red (clean) and green (merged). (b) Zoomed-in view showing the classification results for an output score $CNN_{score} > 0.7$. Note that a logarithmic scale is chosen for the y-axis in (a) while (b) uses a linear scale.

Appendix B

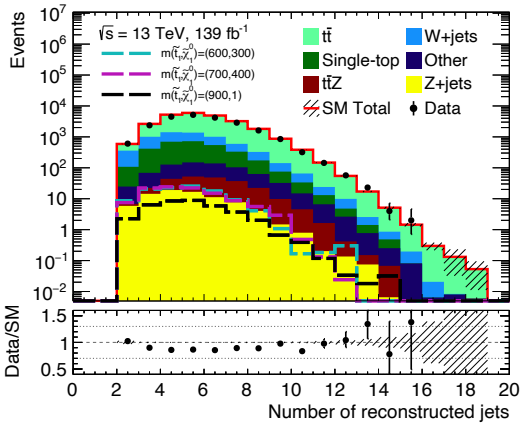
Appendix B

B.1 ATLAS Run 2 GRLs used in tc + $E_{\mathrm{T}}^{\mathrm{miss}}$ Analysis

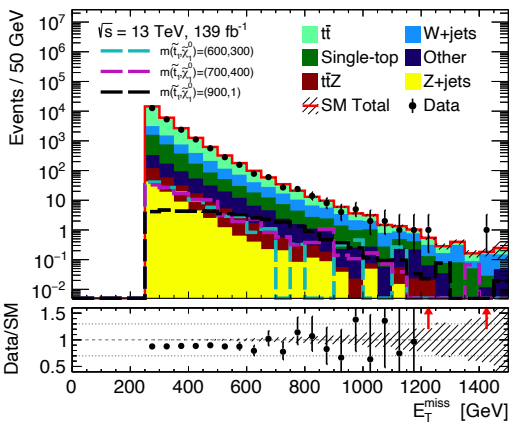
GRL
GoodRunsLists/data15_13TeV/20170619/physics_25ns_21.0.19.xml
GoodRunsLists/data16_13TeV/20180129/physics_25ns_21.0.19.xml
GoodRunsLists/data17_13TeV/20180619/physics_25ns_Triggerno17e33prim.xml
GoodRunsLists/data18_13TeV/20190318/physics_25ns_Triggerno17e33prim.xml

Table B.1: Summary of the GRLs used in the presented tc + $E_{\mathrm{T}}^{\mathrm{miss}}$ analysis.

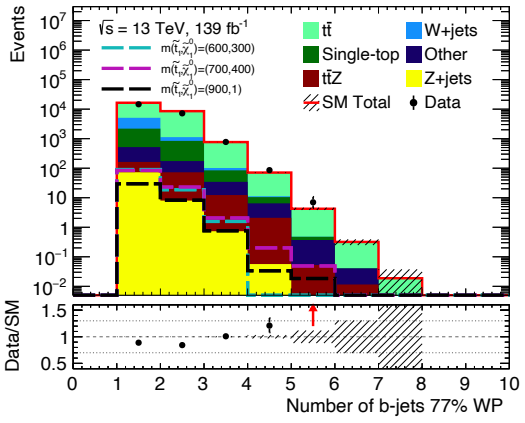
B.2 Common SRA and SRB 1L Preselection



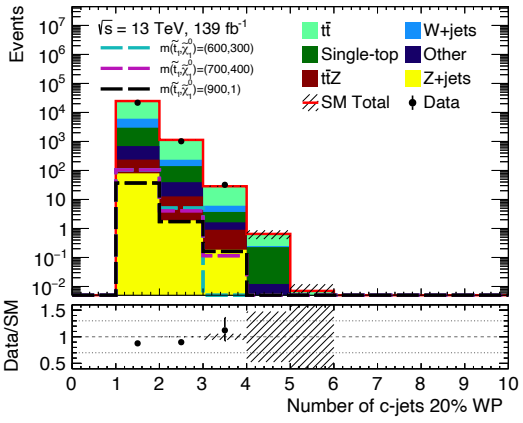
(a) N_{jets}



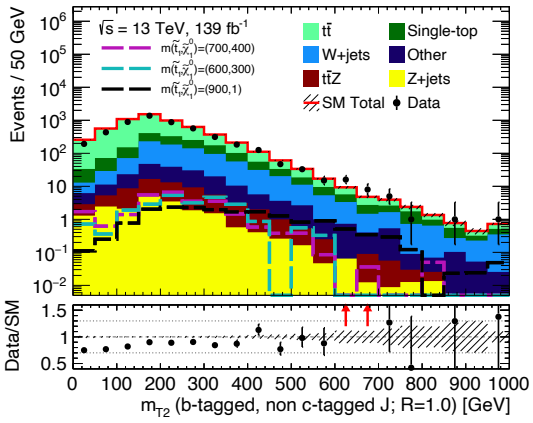
(b) $E_{\text{T}}^{\text{miss}}$



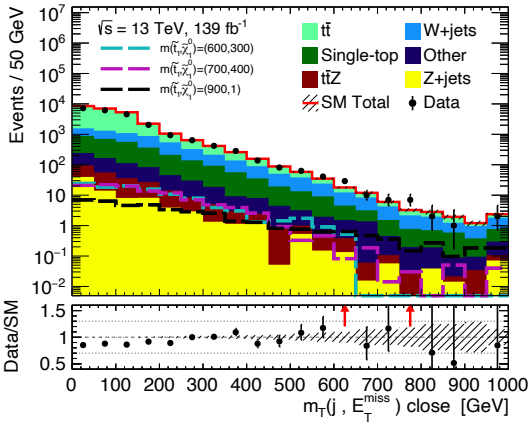
(c) $N_{\text{b-jets}}$



(d) $N_{\text{c-jets}}$



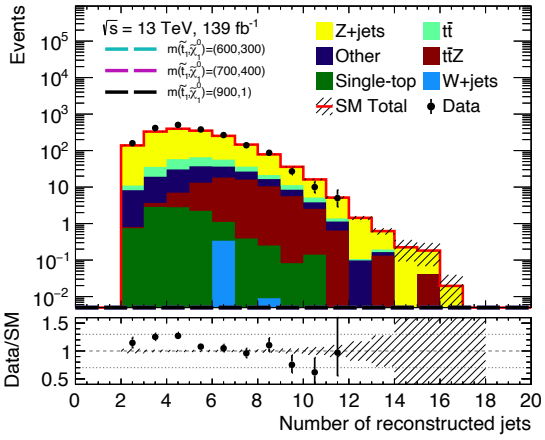
(e) m_{T2}



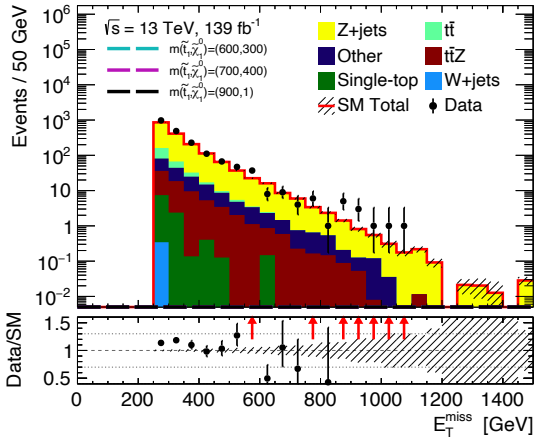
(f) $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$

Figure B.1: Different observables shown at the common SRA and SRB 1L preselection level. Stacked SM background contributions are depicted together with expected signal yields for three different signal scenarios (dashed lines). Data is shown in areas where no significant excess in signal is expected. A ratio between data and SM background predictions is shown in the bottom panel, where statistical uncertainties are indicated by shaded error bands.

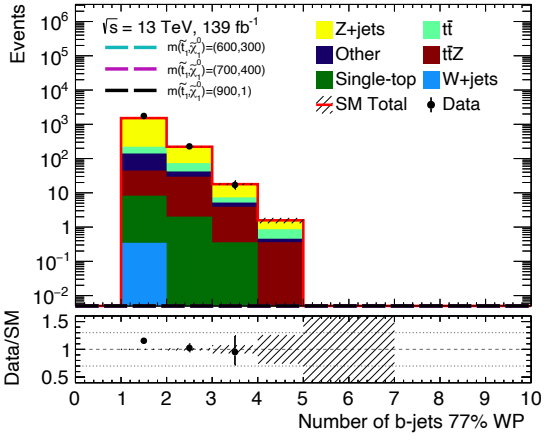
B.3 Common SRA and SRB 2L Preselection



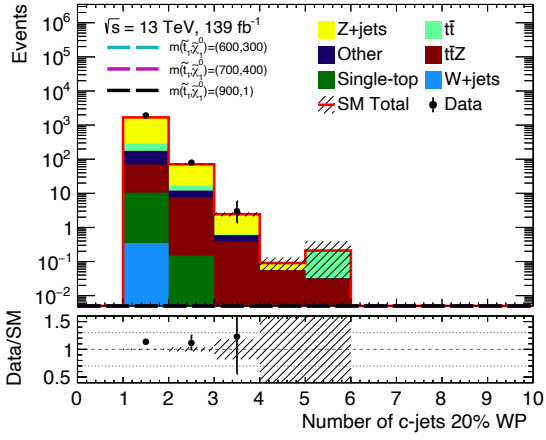
(a) N_{jets}



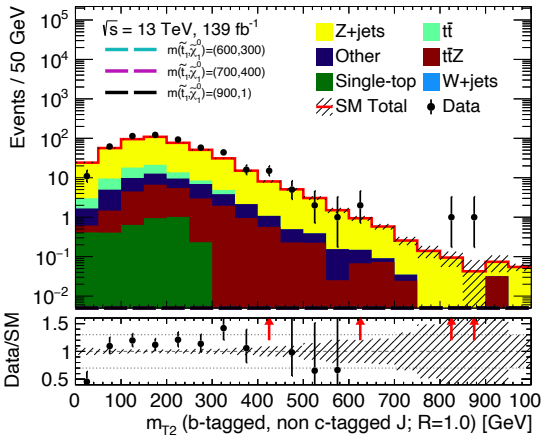
(b) $E_{\text{T}}^{\text{miss}}$



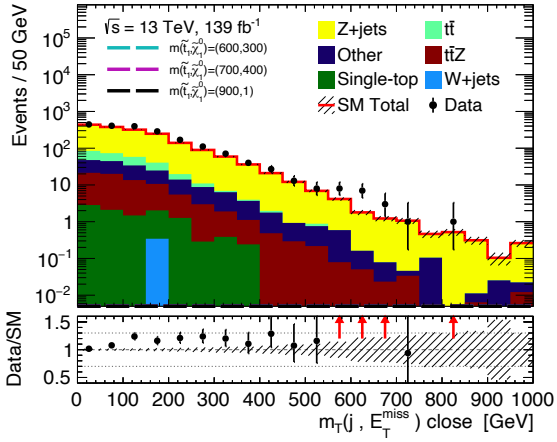
(c) $N_{\text{b-jets}}$



(d) $N_{\text{c-jets}}$



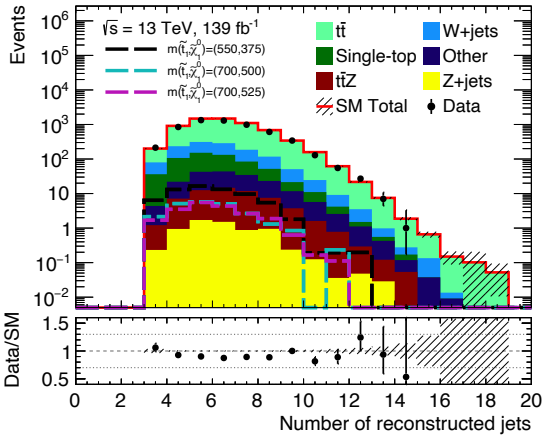
(e) m_{T2}



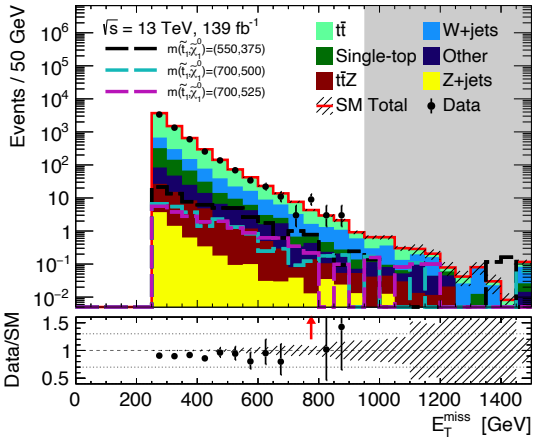
(f) $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$

Figure B.2: Different observables shown at the common SRA and SRB 2L preselection level. Stacked SM background contributions are depicted together with data. A ratio between data and SM background predictions is shown in the bottom panel, where statistical uncertainties are indicated by shaded error bands.

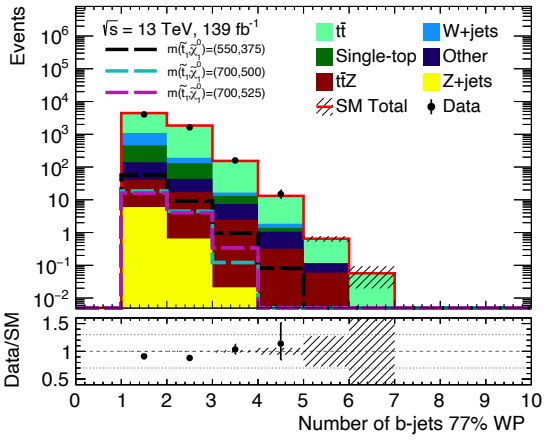
B.4 SRC 1L Preselection



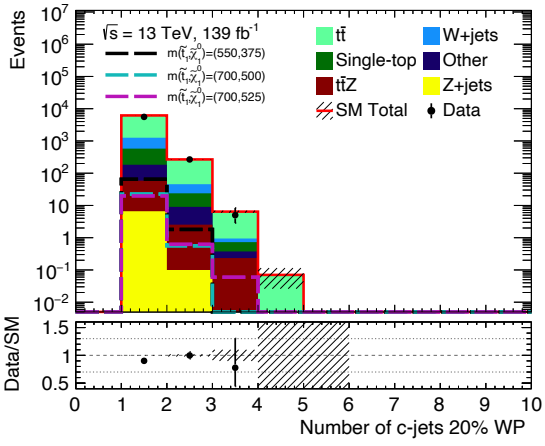
(a) N_{jets}



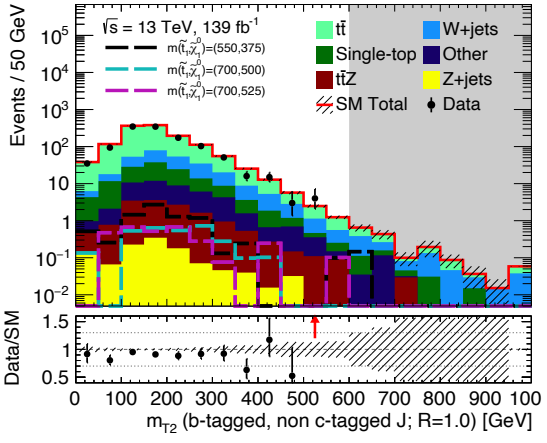
(b) $E_{\text{T}}^{\text{miss}}$



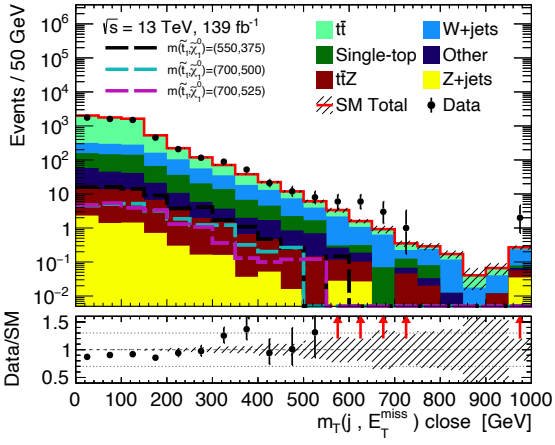
(c) $N_{\text{b-jets}}$



(d) $N_{\text{c-jets}}$



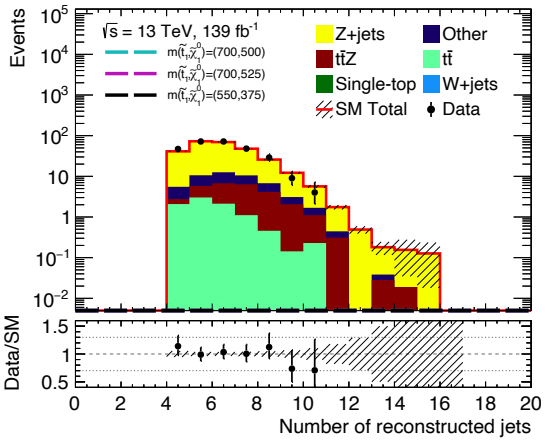
(e) m_{T2}



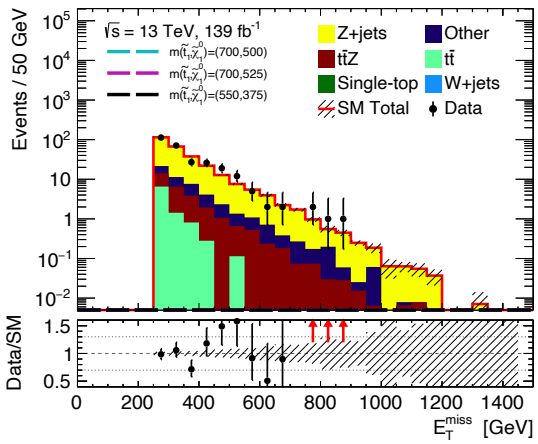
(f) $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$

Figure B.3: Different observables shown at the SRC 1L preselection level. Stacked SM background contributions are depicted together with data. A ratio between data and SM background predictions is shown in the bottom panel, where statistical uncertainties are indicated by shaded error bands.

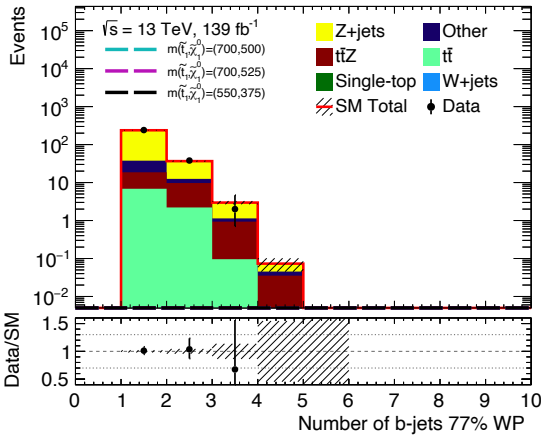
B.5 SRC 2L Preselection



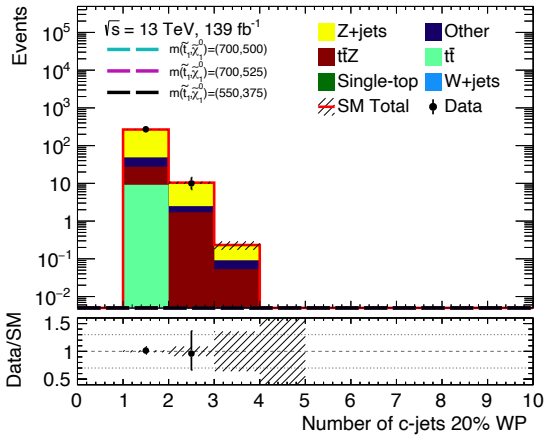
(a) N_{jets}



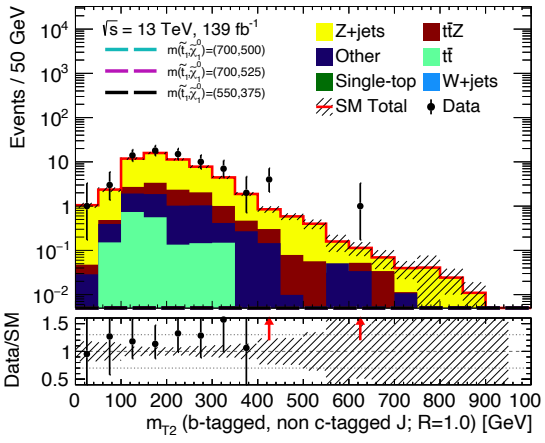
(b) $E_{\text{T}}^{\text{miss}}$



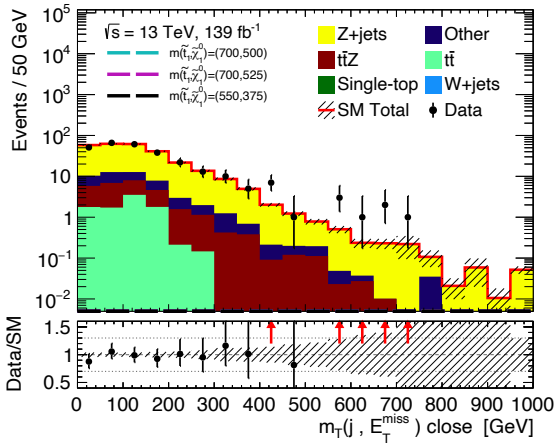
(c) $N_{\text{b-jets}}$



(d) $N_{\text{c-jets}}$



(e) m_{T2}



(f) $m_{\text{T}}(p_{\text{T}}^{\text{j}}, E_{\text{T}}^{\text{miss}})_{\text{close}}$

Figure B.4: Different observables shown at the SRC 2L preselection level. Stacked SM background contributions are depicted together with data. A ratio between data and SM background predictions is shown in the bottom panel, where statistical uncertainties are indicated by shaded error bands.

B.6 VRABZ Signal Contamination

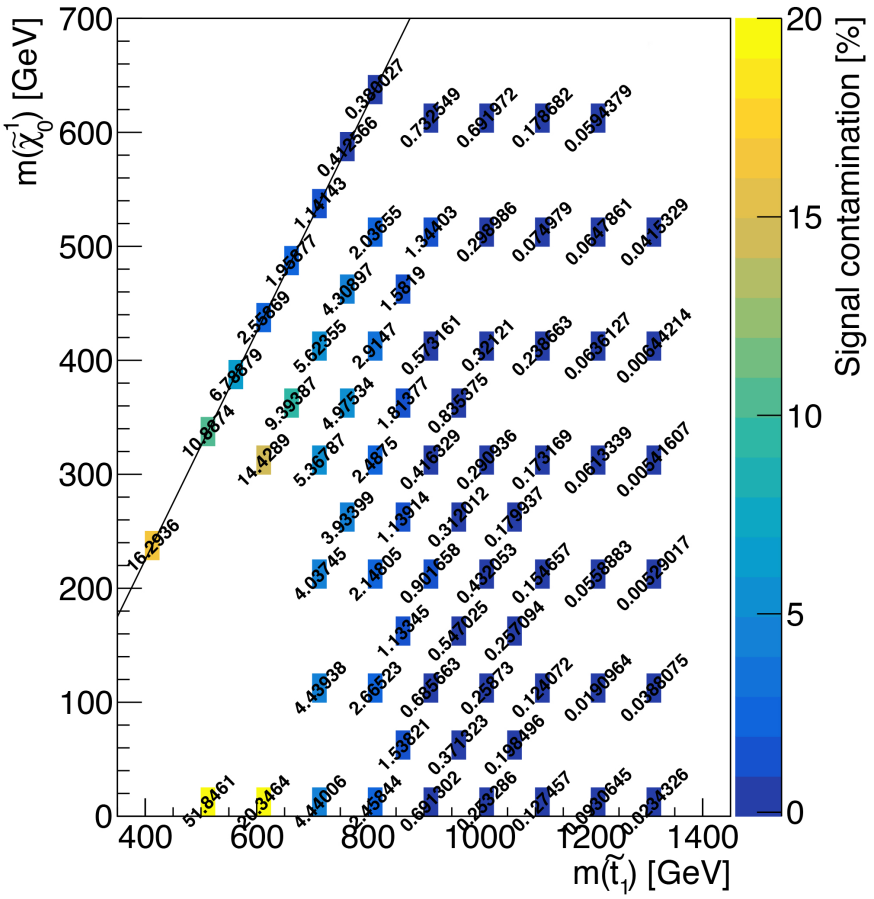


Figure B.5: Signal contamination in VRABZ for different signal points in the mass parameter plane.

B.7 Details on SHERPA Z+Heavy-Flavor Jets Events in Relevant Regions

This section briefly discusses some of the studies that have been conducted to ensure an adequate modeling of the Z+jets background in the SRs.

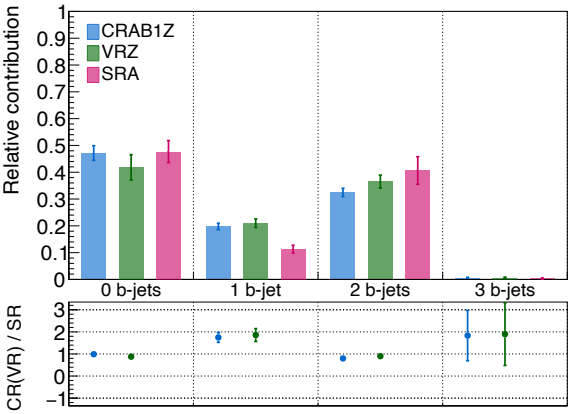
B.7.1 Comparison of Z+Jets Flavor Compositions in SRs, CRs and VRs

Fig. B.6 shows the number of true b- and c-jets in Z+jets events, found in A- and B-type SRs as well as in their respective CRs and VR. An overall good agreement with only small deviations in the number of true b- and c-jet Z+jets events between all shown SRs and their respective CRs and VR can be seen, indicating that events with very similar heavy-flavor jet compositions enter the SRs and their respective CRs and VR used to control and validate the background estimation strategy. In case slightly larger deviations are observed, as for example seen in SRB1 for events with two observed true c-jets, often only the VR contribution slightly deviates from those in the SR and CR, while the contributions in the SR and CR are still very similar.
The truth flavors of the leading b- and c-jets in Z+jets events entering the A- and B-type SRs and their respective CRs and VR is presented in Fig. B.7. Comparing the relative contributions of true light-, c- and b-jets to the leading b- and c-jets in SR, CR and VR, an overall very good agreement with only small deviations is observed, again indicating that similar events in terms of jet flavor composition enter the SRs and their respective CRs and VR.

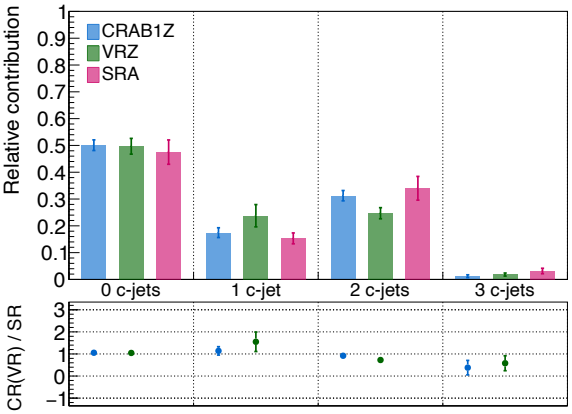
B.7.2 Investigations of Z+c$\bar{c}$b$\bar{b}$ Modeling

In order to better understand if the SHERPA Z+c$\bar{c}$b$\bar{b}$ modeling is adequate, a dedicated region, enriched in Z+c$\bar{c}$b$\bar{b}$ events and free of signal contamination, was optimized to compare and evaluate the agreement between data and pre-fit simulation. Its definition is listed in Table B.2.
While the SRAB 2L preselection ensures the absence of signal contamination, the requirements on m_{ll} and $\Delta R(l_1, l_2)$ are imposed to increase the purity of Z+c$\bar{c}$b$\bar{b}$ events and to decrease the contamination from t$\bar{t}$ events, respectively. Requirements on the number of heavy-flavor jets were found to be the most powerful to discriminate between Z+c$\bar{c}$b$\bar{b}$ events and those Z+jets events not containing a c$\bar{c}$b$\bar{b}$ component (referred to as Z+other jets in the following). Several different combinations of c- and b-tag requirements have been studied[1], the one leading to the highest purity of about 37% Z+c$\bar{c}$b$\bar{b}$ events, while retaining enough statistics, was found to be a $N_{\text{b-jets}} \geq 3$ requirement.
The resulting pre-fit yields of all relevant processes are shown in Table B.3. While Z+c$\bar{c}$b$\bar{b}$ events are the most dominant ones, also a significant amount of Z+other jets as well as $t\bar{t}$Z events enter this region. The overall MC prediction underestimates the observed data by about 15%.

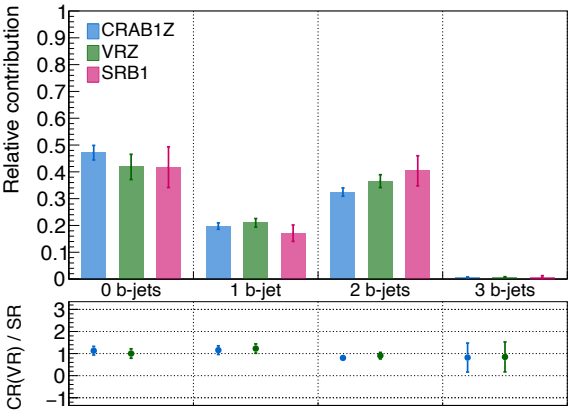
[1] Tested heavy-flavor combinations are: 1b2c, 2b1c, 2b2c, 3b, 3b1c, 3c



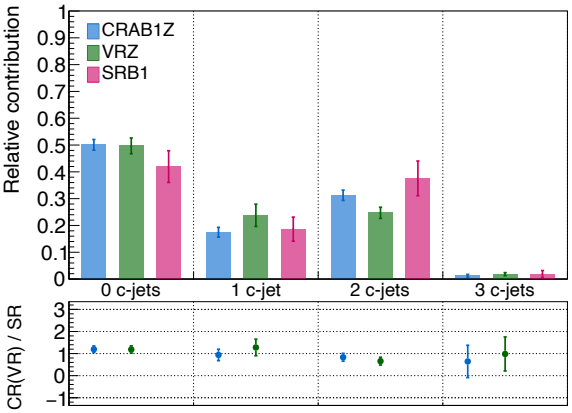
(a) SRA – Number of true b-jets



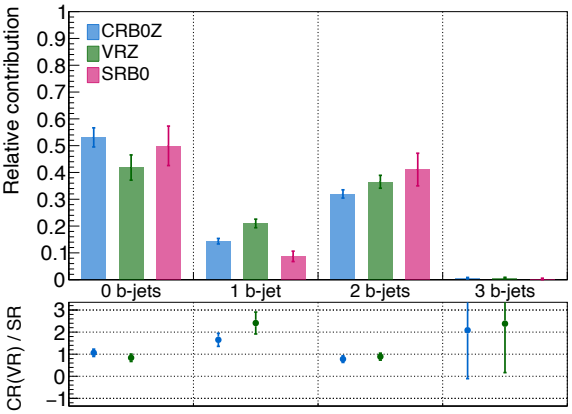
(b) SRA – Number of true c-jets



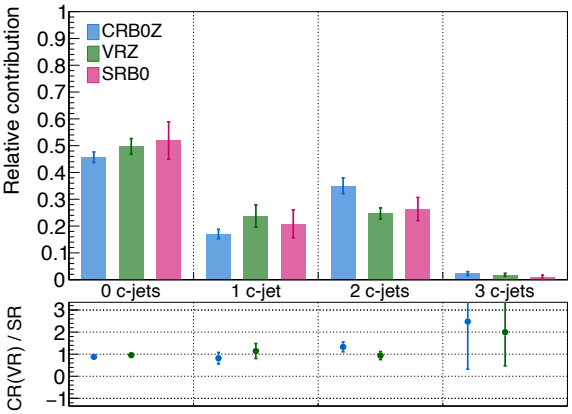
(c) SRB1 – Number of true b-jets



(d) SRB1 – Number of true c-jets

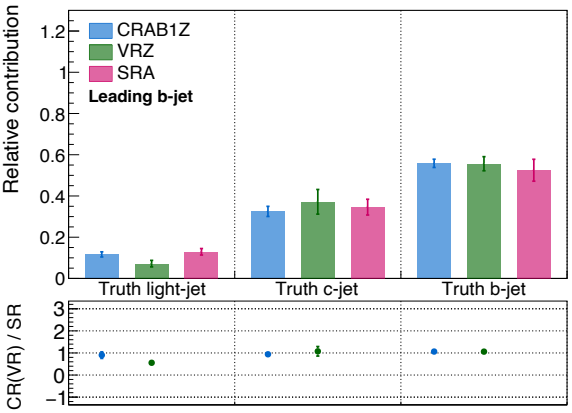


(e) SRB0 – Number of true b-jets

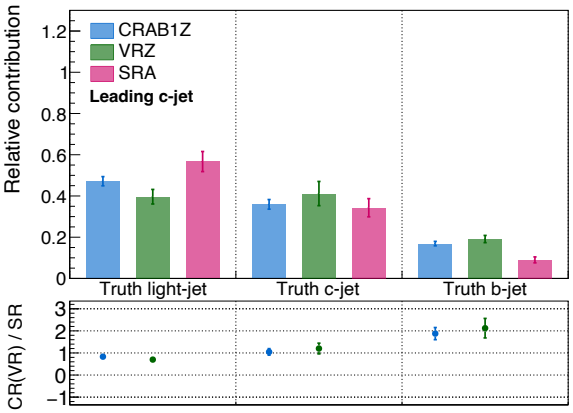


(f) SRB0 – Number of true c-jets

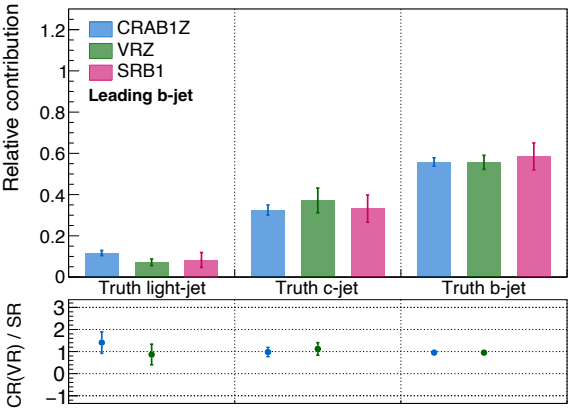
Figure B.6: Number of true b- and c-jets in Z+jets events, found in SRA, SRB1 and SRB0 as well as in their respective CRs and VRs, determined by evaluating and summing the underlying truth flavors of all reconstructed jets in the event. The bottom panel shows the ratios of the CR and VR yields with respect to the yield in the respective SR.



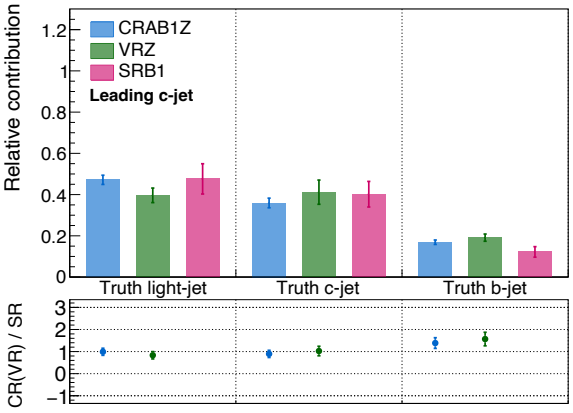
(a) SRA – Leading b-jet truth flavor



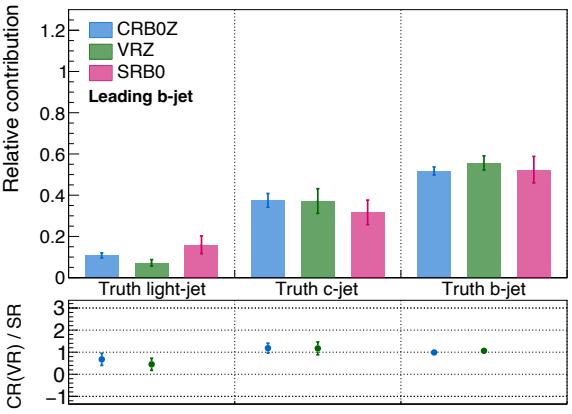
(b) SRA – Leading c-jet truth flavor



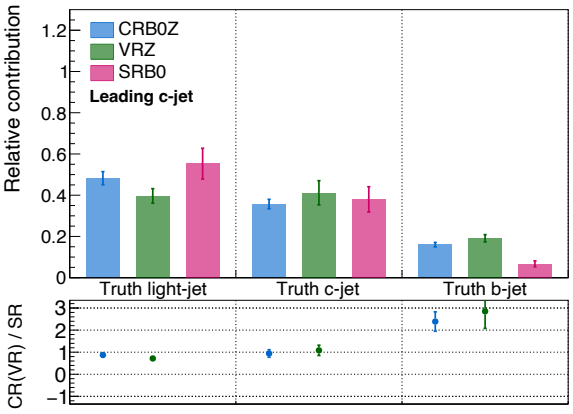
(c) SRB1 – Leading b-jet truth flavor



(d) SRB1 – Leading c-jet truth flavor



(e) SRB0 – Leading b-jet truth flavor



(f) SRB0 – Leading c-jet truth flavor

Figure B.7: Truth flavors of the leading b- and c-jets in Z+jets events, shown for SRA, SRB1 and SRB0 together with their respective CRs and VRs. The bottom panel shows the ratios of the CR and VR yields with respect to the yield in the respective SR.

Kinematic distributions of some selected variables are presented in Fig. B.8, again showing the slight underestimation of data, but no signs of a general major systematic MC mismodeling.

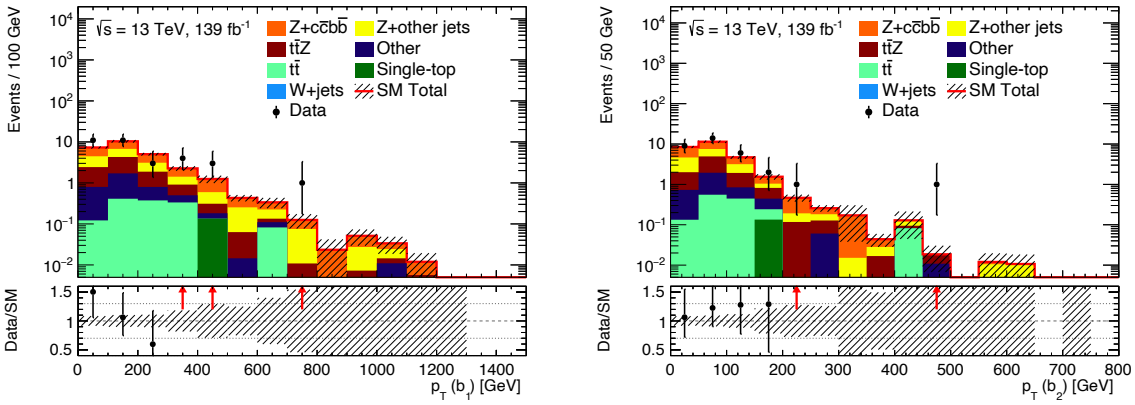
Given the results found in this Z+c$\bar{c}$b$\bar{b}$ enriched region, it can be inferred that an increment of the number of Z+c$\bar{c}$b$\bar{b}$ events by 107% is still compatible with the 28 observed data events within a 1σ discrepancy. It was therefore decided to introduce an additional, conservative 107% systematic uncertainty on events containing Z+c$\bar{c}$b$\bar{b}$ processes in all SRs after the fit, which should cover for potential mismodeling not taken into account by the usual theory systematics, such as the factorization and renormalization scale uncertainties. The above presented flavor composition results and the inclusion of the extra Z+c$\bar{c}$b$\bar{b}$ uncertainty increases the overall confidence that the Z+jets background can be well controlled and validated using the defined CRs and VR, and its uncertainties are properly assessed.

Variable	**Selection**
Preselection	SRAB 2L preselection
m_{ll} [GeV]	$[81, 101]$
$\Delta R(l_1, l_2)$	< 1.3
$N_{\text{b-jets}}$	≥ 3
$p_{\text{T}}^{b_3}$ [GeV]	> 30

Table B.2: Definition of a dedicated Z+c$\bar{c}$b$\bar{b}$ enriched region on top of the common SRAB 2L preselection as defined in Table 9.2.

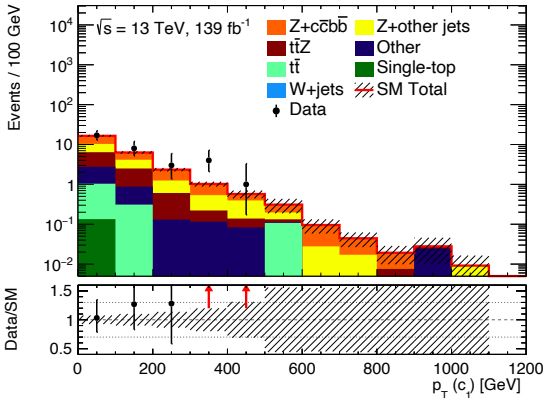
Process	**Yield**
Z+c$\bar{c}$b$\bar{b}$	8.80 ± 0.46
Z+other jets	6.05 ± 0.57
$t\bar{t}$Z	5.37 ± 0.26
Other	2.46 ± 0.45
$t\bar{t}$	1.05 ± 0.36
Single-top	0.13 ± 0.13
W+jets	0.00 ± 0.00
SM	23.87 ± 0.98
Data	28

Table B.3: Pre-fit yields of all relevant processes in the Z+c$\bar{c}$b$\bar{b}$ enriched region. Z+jets processes are split up into a Z+c$\bar{c}$b$\bar{b}$ and Z+other jets component. A purity of about 37% Z+c$\bar{c}$b$\bar{b}$ events is achieved.

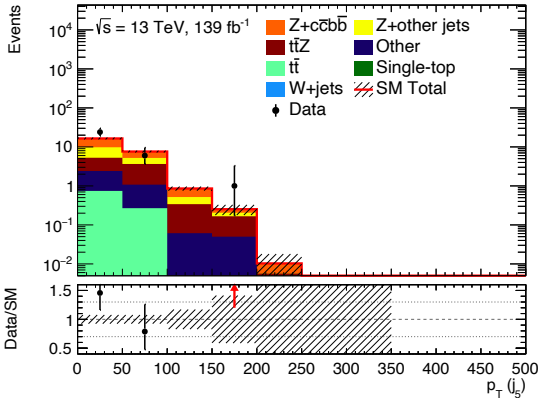


(a) $p_{\mathrm{T}}^{b_1}$

(b) $p_{\mathrm{T}}^{b_2}$



(c) $p_{\mathrm{T}}^{c_1}$



(d) $p_{\mathrm{T}}^{j_5}$

Figure B.8: Distributions of (a) the leading b-jet p_{T}, (b) the sub-leading b-jet p_{T}, (c) the leading c-jet p_{T} and (d) the fifth-leading jet p_{T} in the dedicated Z+c$\bar{\mathrm{c}}$b$\bar{\mathrm{b}}$ enriched region. Only statistical uncertainties are shown.

B.8 Comparison of Single-Top DR/DS Schemes

Fig. B.9 shows the m_{eff} distributions for single-top backgrounds at SRAB and SRC preselection level, comparing the nominal single-top samples, in which the W+t modeling makes use of the diagram removal scheme (DR), and the alternative samples with diagram subtraction scheme (DS). A significant difference between the DR and DS schemes can be observed, which can be identified as the dominant source of systematic uncertainty in single-top processes. The same behavior can be seen in Table B.4, listing the SRAB and SRC preselection yields for single-top processes with DR and DS schemes.

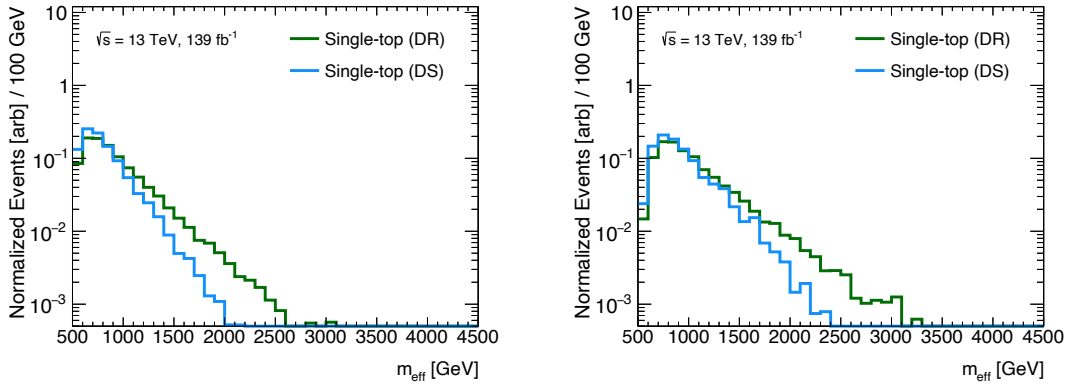


Figure B.9: Comparisons of DR and DS single-top schemes for (a) SRAB preselection and (b) SRC preselection, showing the m_{eff} distributions.

Process	**SRAB preselection**	**SRC preselection**
Single-top (DR)	1360.29 ± 10.60	593.76 ± 6.86
Single-top (DS)	734.15 ± 6.52	323.03 ± 4.06

Table B.4: SRAB and SRC preselection yields for single-top (DR) and single-top (DS) backgrounds. Only statistical uncertainties are shown.

B.9 Systematic Tables – SRC

Region	SRC-750	SRC-1000	SRC-1250
Total background expectation	369.38	202.81	55.60
Total statistical uncertainty	±19.22	±14.24	±7.46
Total systematic uncertainty	±119.76 [32.42%]	±66.79 [32.93%]	±18.70 [33.64%]
α_ctag_weight	±110.81 [**30.0%**]	±60.84 [**30.0%**]	±16.68 [**30.0%**]
μ_ttbar	±26.92 [**7.3%**]	±15.84 [**7.8%**]	±7.22 [**13.0%**]
μ_W	±21.59 [**5.8%**]	±13.77 [**6.8%**]	±3.68 [**6.6%**]
α_JER_EffectiveNP_7restTerm	±18.96 [5.1%]	±2.47 [1.2%]	±1.05 [1.9%]
α_ttbar_theory_GEN	±15.44 [4.2%]	±2.11 [1.0%]	±0.90 [1.6%]
α_JER_EffectiveNP_6	±15.42 [4.2%]	±3.32 [1.6%]	±0.04 [0.08%]
α_Zccbb	±12.02 [3.3%]	±6.52 [3.2%]	±2.36 [4.2%]
α_JER_MC16	±11.47 [3.1%]	±6.69 [3.3%]	±0.47 [0.85%]
α_FT_EFF_B_1_systematics	±9.95 [2.7%]	±4.37 [2.2%]	±0.87 [1.6%]
α_st_theory_PS	±9.74 [2.6%]	±3.78 [1.9%]	±1.78 [3.2%]
α_JER_EffectiveNP_2	±7.26 [2.0%]	±2.15 [1.1%]	±0.18 [0.33%]
α_ttbar_theory_ISR	±6.92 [1.9%]	±1.12 [0.55%]	±0.52 [0.94%]
α_JER_EffectiveNP_1	±6.54 [1.8%]	±1.42 [0.70%]	±0.05 [0.09%]
μ_Z_C	±5.99 [1.6%]	±3.74 [1.8%]	±1.08 [1.9%]
α_JER_EffectiveNP_4	±5.30 [1.4%]	±4.43 [2.2%]	±0.48 [0.86%]
α_st_theory_DS	±4.20 [1.1%]	±3.34 [1.6%]	±1.19 [2.1%]
α_FT_EFF_B_2_systematics	±3.93 [1.1%]	±2.36 [1.2%]	±0.63 [1.1%]
α_EL_EFF_ID	±3.20 [0.87%]	±1.59 [0.78%]	±0.41 [0.74%]
α_JER_EffectiveNP_5	±2.71 [0.73%]	±1.43 [0.70%]	±0.11 [0.20%]
α_st_theory_GEN	±2.69 [0.73%]	±4.36 [2.1%]	±1.36 [2.5%]
α_FT_EFF_extrapolation	±2.66 [0.72%]	±1.34 [0.66%]	±0.44 [0.79%]
α_ttbar_theory_PS	±2.28 [0.62%]	±0.70 [0.35%]	±3.15 [5.7%]
α_JER_EffectiveNP_3	±2.23 [0.60%]	±5.22 [2.6%]	±0.05 [0.10%]
α_FT_EFF_B_3_systematics	±2.23 [0.60%]	±0.80 [0.39%]	±0.11 [0.21%]
α_FT_EFF_extrapolation	±2.07 [0.56%]	±0.10 [0.05%]	±0.20 [0.37%]
α_ttbar_theory_FSR	±2.02 [0.55%]	±0.07 [0.03%]	±0.35 [0.63%]
α_JET_Flavor_Response	±1.66 [0.45%]	±1.93 [0.95%]	±0.89 [1.6%]
α_JET_JvtEfficiency	±1.47 [0.40%]	±0.66 [0.33%]	±0.14 [0.25%]
α_FT_EFF_L_1_systematics	±1.46 [0.40%]	±0.76 [0.38%]	±0.27 [0.49%]
α_JET_GroupedNP_1	±1.44 [0.39%]	±3.73 [1.8%]	±0.60 [1.1%]
α_PDF_CT14	±1.44 [0.39%]	±1.14 [0.57%]	±0.29 [0.52%]
α_PDF_MMHT14	±1.44 [0.39%]	±1.15 [0.57%]	±0.27 [0.49%]
α_ttbar_theory_μR_μF	±1.27 [0.34%]	±0.75 [0.37%]	±0.75 [0.37%]
γ_stat_SRC_cuts	±1.26 [0.34%]	±0.73 [0.36%]	±0.75 [0.37%]

Table B.5: Breakdown of the dominant systematic uncertainties in the first three SRC m_{eff} bins after the background-only fit. Individual uncertainties can be correlated and do not necessarily add up in quadrature. The three most dominant systematic uncertainties in each region are shown in bold.

Region	SRC-1500	SRC-1750	SRC-2000
Total background expectation	20.83	6.99	3.40
Total statistical uncertainty	±4.56	±2.64	±1.84
Total systematic uncertainty	±7.92 [38.00%]	±2.51 [35.90%]	±1.45 [42.53%]
α_ctag_weight	±6.24 [**30.0%**]	±2.10 [**30.0%**]	±1.02 [**30.0%**]
μ_ttbar	±2.52 [**12.1%**]	±0.94 [**13.4%**]	±0.49 [**14.3%**]
α_MET_SoftTrk_ResoPerp	±1.70 [**8.2%**]	±0.16 [2.3%]	±0.06 [1.7%]
α_st_theory_PS	±1.41 [6.8%]	±0.03 [0.50%]	±0.25 [7.4%]
α_JER_EffectiveNP_5	±1.41 [6.8%]	±0.23 [3.3%]	±0.11 [3.2%]
μ_W	±1.38 [6.6%]	±0.48 [6.8%]	±0.21 [6.2%]
α_JER_EffectiveNP_3	±1.26 [6.0%]	±0.07 [1.0%]	±0.11 [3.2%]
α_JER_EffectiveNP_1	±1.17 [5.6%]	±0.22 [3.1%]	±0.03 [0.86%]
α_JER_EffectiveNP_2	±1.13 [5.4%]	±0.06 [0.88%]	±0.03 [0.92%]
α_JER_EffectiveNP_4	±1.04 [5.0%]	±0.14 [2.0%]	±0.02 [0.64%]
α_Zccbb	±1.01 [4.9%]	±0.22 [3.1%]	±0.11 [3.4%]
α_ttbar_theory_GEN	±0.89 [4.3%]	±0.18 [2.5%]	±0.01 [0.19%]
α_st_theory_DS	±0.86 [4.1%]	±0.29 [4.2%]	±0.19 [5.5%]
α_MET_SoftTrk_ResoPara	±0.85 [4.1%]	±0.25 [3.6%]	±0.05 [1.4%]
α_ttbar_theory_PS	±0.65 [3.1%]	±0.26 [3.7%]	±0.52 [**15.2%**]
α_ttbar_theory_ISR	±0.55 [2.6%]	±0.01 [0.18%]	±0.22 [6.4%]
α_JET_Flavor_Response	±0.53 [2.5%]	±0.15 [2.1%]	±0.12 [3.6%]
μ_Z_C	±0.45 [2.1%]	±0.12 [1.7%]	±0.06 [1.7%]
α_JET_GroupedNP_3	±0.44 [2.1%]	±0.14 [2.0%]	±0.14 [2.0%]
γ_stat_SRC_cuts	±0.43 [2.0%]	±0.13 [1.9%]	±0.14 [2.0%]
α_JER_EffectiveNP_7restTerm	±0.37 [1.8%]	±0.22 [3.1%]	±0.10 [3.0%]
α_PILEUP	±0.27 [1.3%]	±0.03 [0.47%]	±0.02 [0.64%]
α_FT_EFF_B_1_systematics	±0.27 [1.3%]	±0.10 [1.4%]	±0.04 [1.2%]
α_JET_GroupedNP_1	±0.26 [1.2%]	±0.13 [1.8%]	±0.06 [1.8%]
α_ttbar_theory_FSR	±0.24 [1.2%]	±0.09 [1.3%]	±0.03 [0.83%]
α_FT_EFF_B_2_systematics	±0.24 [1.1%]	±0.08 [1.1%]	±0.04 [1.3%]
α_MET_SoftTrk_Scale	±0.23 [1.1%]	±0.03 [0.48%]	±0.01 [0.29%]
α_JER_MC16	±0.23 [1.1%]	±0.44 [6.3%]	±0.31 [9.1%]
α_JET_GroupedNP_2	±0.23 [1.1%]	±0.10 [1.4%]	±0.02 [0.70%]
α_FT_EFF_extrapolation	±0.20 [0.98%]	±0.07 [0.98%]	±0.01 [0.41%]
α_JER_EffectiveNP_6	±0.18 [0.87%]	±0.33 [4.7%]	±0.26 [7.8%]
α_st_theory_GEN	±0.17 [0.84%]	±0.52 [**7.5%**]	±0.15 [4.4%]
α_EL_EFF_ID	±0.16 [0.75%]	±0.05 [0.72%]	±0.02 [0.73%]
α_FT_EFF_L_1_systematics	±0.12 [0.59%]	±0.04 [0.56%]	±0.01 [0.39%]
α_W_theory_ckkw	±0.09 [0.43%]	±0.05 [0.67%]	±0.01 [0.33%]
α_ttbar_theory_μR_μF	±0.09 [0.42%]	±0.16 [2.2%]	±0.14 [4.1%]
α_PDF_CT14	±0.08 [0.37%]	±0.05 [0.68%]	±0.02 [0.64%]
α_PDF_MMHT14	±0.07 [0.33%]	±0.05 [0.68%]	±0.02 [0.63%]

Table B.6: Breakdown of the dominant systematic uncertainties in the last three SRC m_{eff} bins after the background-only fit. Individual uncertainties can be correlated and do not necessarily add up in quadrature. The three most dominant systematic uncertainties in each region are shown in bold.

Bibliography

[1] Steven Weinberg. "The Making of the Standard Model". In: (2003). URL: http://cds.cern.ch/record/799984/files/0401010.pdf?version=1.

[2] Mary K. Gaillard, Paul D. Grannis, and Frank J. Sciulli. "The Standard Model of Particle Physics". In: *Reviews of Modern Physics* 71.2 (1999), S96–S111. ISSN: 1539-0756. DOI: 10.1103/revmodphys.71.s96. URL: http://dx.doi.org/10.1103/RevModPhys.71.S96.

[3] Cush MissMJ. *Standard Model of Elementary Particles*. https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg. 2021.

[4] Michael Peskin and Dan Schroeder. *An Introduction to Quantum Field Theory*. Westview Press, 1995.

[5] Alexandre Deur, Stanley J. Brodsky, and Guy F. de Téramond. "The QCD running coupling". In: *Progress in Particle and Nuclear Physics* 90 (2016), 1–74. ISSN: 0146-6410. DOI: 10.1016/j.ppnp.2016.04.003. URL: http://dx.doi.org/10.1016/j.ppnp.2016.04.003.

[6] H David Politzer. "Asymptotic freedom: An approach to strong interactions". In: *Physics Reports* 14.4 (1974), pp. 129–180. ISSN: 0370-1573. DOI: https://doi.org/10.1016/0370-1573(74)90014-3. URL: https://www.sciencedirect.com/science/article/pii/0370157374900143.

[7] Bo Andersson, Sandipan Mohanty, and Fredrik Soderberg. "Recent developments in the Lund model". In: *36th Annual Winter School on Nuclear and Particle Physics (PINP 2002) and 8th St. Petersburg School on Theoretical Physics St. Petersburg, Russia, February 25-March 3, 2002*. 2002. arXiv: hep-ph/0212122 [hep-ph].

[8] Sheldon L. Glashow. "Partial-symmetries of weak interactions". In: *Nuclear Physics* 22.4 (1961), pp. 579–588. ISSN: 0029-5582. DOI: https://doi.org/10.1016/0029-5582(61)90469-2. URL: https://www.sciencedirect.com/science/article/pii/0029558261904692.

[9] Steven Weinberg. "A Model of Leptons". In: *Phys. Rev. Lett.* 19 (21 1967), pp. 1264–1266. DOI: 10.1103/PhysRevLett.19.1264. URL: https://link.aps.org/doi/10.1103/PhysRevLett.19.1264.

[10] A Salam. "WEAK AND ELECTROMAGNETIC INTERACTIONS." In: *pp 367-77 of Elementary Particle Theory. Svartholm, Nils (ed.). New York, John Wiley and Sons, Inc., 1968.* (Oct. 1969). URL: https://www.osti.gov/biblio/4767615.

[11] Yoichiro Nambu. "Quasi-Particles and Gauge Invariance in the Theory of Superconductivity". In: *Phys. Rev.* 117 (3 1960), pp. 648–663. DOI: 10.1103/PhysRev.117.648. URL: https://link.aps.org/doi/10.1103/PhysRev.117.648.

[12] J Goldstone. "Field theories with "superconductor" solutions". In: *Nuovo Cimento* 19 (1960), pp. 154–164. DOI: 10.1007/BF02812722. URL: http://cds.cern.ch/record/343400.

[13] F. Englert and R. Brout. "Broken Symmetry and the Mass of Gauge Vector Mesons". In: *Phys. Rev. Lett.* 13 (9 1964), pp. 321–323. DOI: 10.1103/PhysRevLett.13.321. URL: https://link.aps.org/doi/10.1103/PhysRevLett.13.321.

[14] P.W. Higgs. "Broken symmetries, massless particles and gauge fields". In: *Physics Letters* 12.2 (1964), pp. 132–133. ISSN: 0031-9163. DOI: https://doi.org/10.1016/0031-9163(64)91136-9. URL: https://www.sciencedirect.com/science/article/pii/0031916364911369.

[15] G. Aad et al. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Physics Letters B* 716.1 (2012), 1–29. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.020. URL: http://dx.doi.org/10.1016/j.physletb.2012.08.020.

[16] S. Chatrchyan et al. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Physics Letters B* 716.1 (2012), 30–61. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.08.021. URL: http://dx.doi.org/10.1016/j.physletb.2012.08.021.

[17] Gianfranco Bertone, Dan Hooper, and Joseph Silk. "Particle dark matter: evidence, candidates and constraints". In: *Physics Reports* 405.5-6 (2005), 279–390. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2004.08.031. URL: http://dx.doi.org/10.1016/j.physrep.2004.08.031.

[18] F. Zwicky. "Republication of: The redshift of extragalactic nebulae". In: *General Relativity and Gravitation* 41.1 (Jan. 2009), pp. 207–224. DOI: 10.1007/s10714-008-0707-4.

[19] Vera C. Rubin and Jr. Ford W. Kent. "Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions". In: 159 (Feb. 1970), p. 379. DOI: 10.1086/150317.

[20] Douglas Clowe et al. "A Direct Empirical Proof of the Existence of Dark Matter". In: *The Astrophysical Journal* 648.2 (2006), L109–L113. ISSN: 1538-4357. DOI: 10.1086/508162. URL: http://dx.doi.org/10.1086/508162.

[21] P. A. R. Ade et al. "Planck2015 results". In: 594 (2016), A13. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201525830. URL: http://dx.doi.org/10.1051/0004-6361/201525830.

[22] E. Komatsu et al. "Five-year Wilkinson mircowave anisotroopy probe observations: Cosmological interpreation". In: *The Astrophysical Journal Supplement Series* 180.2 (2009), 330–376. ISSN: 1538-4365. DOI: 10.1088/0067-0049/180/2/330. URL: http://dx.doi.org/10.1088/0067-0049/180/2/330.

[23] A. D. Sakharov. "Violation of CP Invariance, C Asymmetry, and Baryon Asymmetry of the Universe". In: *Soviet Journal of Experimental and Theoretical Physics Letters* 5 (Jan. 1967), p. 24.

[24] Michael Dine and Alexander Kusenko. "Origin of the matter-antimatter asymmetry". In: *Rev. Mod. Phys.* 76 (1 2003), pp. 1–30. DOI: 10.1103/RevModPhys.76.1. URL: https://link.aps.org/doi/10.1103/RevModPhys.76.1.

[25] Ian Aitchison. "Supersymmetry in Particle Physics: An Elementary Introduction". In: *Supersymmetry in Particle Physics: An Elementary Introduction, by Ian J. R. Aitchison. ISBN 978-0-521-88023-7. Published by Cambridge University Press, Cambridge, UK, 2007.* (Jan. 2007). DOI: 10.1017/CBO9780511619250.

[26] H.P. Nilles. "Supersymmetry, supergravity and particle physics". In: *Physics Reports* 110.1 (1984), pp. 1–162. ISSN: 0370-1573. DOI: https://doi.org/10.1016/0370-1573(84)90008-5. URL: https://www.sciencedirect.com/science/article/pii/0370157384900085.

[27] H.E. Haber and G.L. Kane. "The search for supersymmetry: Probing physics beyond the standard model". In: *Physics Reports* 117.2 (1985), pp. 75–263. ISSN: 0370-1573. DOI: https://doi.org/10.1016/0370-1573(85)90051-1. URL: https://www.sciencedirect.com/science/article/pii/0370157385900511.

[28] STEPHEN P. MARTIN. "A SUPERSYMMETRY PRIMER". In: *Advanced Series on Directions in High Energy Physics* (1998), 1–98. ISSN: 1793-1339. DOI: 10.1142/9789812839657_0001. URL: http://dx.doi.org/10.1142/9789812839657_0001.

[29] Ian Aitchison. "Supersymmetry in Particle Physics: An Elementary Introduction". In: *Supersymmetry in Particle Physics: An Elementary Introduction, by Ian J. R. Aitchison. ISBN 978-0-521-88023-7. Published by Cambridge University Press, Cambridge, UK, 2007.* (Jan. 2007). DOI: 10.1017/CBO9780511619250.

[30] CSABA CSÁKI. "THE MINIMAL SUPERSYMMETRIC STANDARD MODEL". In: *Modern Physics Letters A* 11.08 (1996), 599–613. ISSN: 1793-6632. DOI: 10.1142/s021773239600062x. URL: http://dx.doi.org/10.1142/S021773239600062X.

[31] G. D'Ambrosio et al. "Minimal flavour violation: an effective field theory approach". In: *Nuclear Physics B* 645.1 (2002), pp. 155–187. ISSN: 0550-3213. DOI: https://doi.org/10.1016/S0550-3213(02)00836-2. URL: https://www.sciencedirect.com/science/article/pii/S0550321302008362.

[32] Karen De Causmaecker et al. "General squark flavour mixing: constraints, phenomenology and benchmarks". In: *Journal of High Energy Physics* 2015.11 (2015). ISSN: 1029-8479. DOI: 10.1007/jhep11(2015)125. URL: http://dx.doi.org/10.1007/JHEP11(2015)125.

[33] A. Bartl et al. "Impact of squark generation mixing on the search for squarks decaying into fermions at LHC". In: *Physics Letters B* 698.5 (2011), pp. 380–388. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2011.01.020. URL: https://www.sciencedirect.com/science/article/pii/S037026931100044X.

[34] Carola F Berger et al. "Supersymmetry without prejudice". In: *Journal of High Energy Physics* 2009.02 (2009), 023–023. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2009/02/023. URL: http://dx.doi.org/10.1088/1126-6708/2009/02/023.

[35] C. Regis et al. "Search for proton decay via $p \to \mu^+ K^0$ in Super-Kamiokande I, II, and III". In: *Phys. Rev. D* 86 (1 2012), p. 012006. DOI: 10.1103/PhysRevD.86.012006. URL: https://link.aps.org/doi/10.1103/PhysRevD.86.012006.

[36] K. Abe et al. "Search for proton decay via $p \to \nu K^+$ using 260 kiloton · year data of Super-Kamiokande". In: *Phys. Rev. D* 90 (7 2014), p. 072005. DOI: 10.1103/PhysRevD.90.072005. URL: https://link.aps.org/doi/10.1103/PhysRevD.90.072005.

[37] K. Abe et al. "Search for proton decay via $p \to e^+\pi^0$ and $p \to \mu^+\pi^0$ in 0.31 megaton· years exposure of the Super-Kamiokande water Cherenkov detector". In: *Phys. Rev. D* 95 (1 2017), p. 012004. DOI: 10.1103/PhysRevD.95.012004. URL: https://link.aps.org/doi/10.1103/PhysRevD.95.012004.

[38] Glennys R. Farrar and Pierre Fayet. "Phenomenology of the production, decay, and detection of new hadronic states associated with supersymmetry". English (US). In: *Physics Letters, Section B: Nuclear, Elementary Particle and High-Energy Physics* 76.5 (July 1978), pp. 575–579. ISSN: 0370-2693. DOI: 10.1016/0370-2693(78)90858-4.

[39] M. Aaboud et al. "Search for top-squark pair production in final states with one lepton jets and missing transverse momentum using 36 fb-1 of 13 TeV pp collision data with the ATLAS detector". In: *Journal of High Energy Physics* 2018.6 (2018). ISSN: 1029-8479. DOI: 10.1007/jhep06(2018)108. URL: http://dx.doi.org/10.1007/JHEP06(2018)108.

[40] M. Aaboud et al. "Search for supersymmetry in final states with charm jets and missing transverse momentum in 13 TeV pp collisions with the ATLAS detector". In: *Journal of High Energy Physics* 2018.9 (2018). ISSN: 1029-8479. DOI: 10.1007/jhep09(2018)050. URL: http://dx.doi.org/10.1007/JHEP09(2018)050.

[41] Amit Chakraborty et al. "Flavour-violating decays of mixed top-charm squarks at the LHC". In: *The European Physical Journal C* 78.10 (2018). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-018-6331-x. URL: http://dx.doi.org/10.1140/epjc/s10052-018-6331-x.

[42] Lyndon Evans and Philip Bryant. "LHC Machine". In: *Journal of Instrumentation* 3.08 (2008), S08001–S08001. DOI: 10.1088/1748-0221/3/08/s08001. URL: https://doi.org/10.1088/1748-0221/3/08/s08001.

[43] "Linear accelerator 2". In: (2012). URL: https://cds.cern.ch/record/1997427.

[44] J. Vollaire et al. *Linac4 design report*. Ed. by Maurizio Vretenar. Vol. 6. CERN Yellow Reports: Monographs. Geneva: CERN, Sept. 2020. ISBN: 978-92-9083-579-0, 978-92-9083-580-6. DOI: 10.23731/CYRM-2020-006.

[45] "The Proton Synchrotron Booster". In: (2012). URL: https://cds.cern.ch/record/1997372.

[46] "The Proton Synchrotron". In: (2012). URL: https://cds.cern.ch/record/1997189.

[47] "The Super Proton Synchrotron". In: (2012). URL: https://cds.cern.ch/record/1997188.

[48] "Linear accelerator 3". In: (2012). URL: https://cds.cern.ch/record/1997426.

[49] M Chanel. "LEIR: The low energy ion ring at CERN". In: *Nucl. Instrum. Methods Phys. Res., A* 532.CERN-PS-2002-015-AE. CERN-AB-2003-086-ABP (2002), 137–143. 9 p. URL: https://cds.cern.ch/record/557588.

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

[51] The ALICE Collaboration et al. "The ALICE experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08002–S08002. DOI: 10.1088/1748-0221/3/08/s08002. URL: https://doi.org/10.1088/1748-0221/3/08/s08002.

[52] G Aad et al. "The ATLAS Experiment at the CERN Large Hadron Collider". In: *JINST* 3 (2008). Also published by CERN Geneva in 2010, S08003. 437 p. DOI: 10.1088/1748-0221/3/08/S08003. URL: https://cds.cern.ch/record/1129811.

[53] The CMS Collaboration et al. "The CMS experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08004–S08004. DOI: 10.1088/1748-0221/3/08/s08004. URL: https://doi.org/10.1088/1748-0221/3/08/s08004.

[54] The LHCb Collaboration et al. "The LHCb Detector at the LHC". In: *Journal of Instrumentation* 3.08 (2008), S08005–S08005. DOI: 10.1088/1748-0221/3/08/s08005. URL: https://doi.org/10.1088/1748-0221/3/08/s08005.

[55] Werner Herr and B Muratori. *Concept of luminosity*. 2006. DOI: 10.5170/CERN-2006-002.361. URL: https://cds.cern.ch/record/941318.

[56] "ATLAS Luminosity Public Results Run2." In: (). URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2.

[57] 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). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-016-4395-z. URL: http://dx.doi.org/10.1140/epjc/s10052-016-4395-z.

[58] "High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1". In: 4/2017 (2017). Ed. by G. Apollinari et al. DOI: 10.23731/CYRM-2017-004.

[59] Joao Pequenao. "Computer generated image of the whole ATLAS detector". 2008. URL: https://cds.cern.ch/record/1095924.

[60] G Aad et al. "ATLAS pixel detector electronics and sensors". In: *Journal of Instrumentation* 3.07 (2008), P07007–P07007. DOI: 10.1088/1748-0221/3/07/p07007. URL: https://doi.org/10.1088/1748-0221/3/07/p07007.

[61] M Capeans et al. *ATLAS Insertable B-Layer Technical Design Report*. Tech. rep. CERN-LHCC-2010-013. ATLAS-TDR-19. 2010. URL: https://cds.cern.ch/record/1291633.

[62] The ATLAS collaboration. "Operation and performance of the ATLAS semiconductor tracker". In: *Journal of Instrumentation* 9.08 (2014), P08009–P08009. ISSN: 1748-0221. DOI: 10.1088/1748-0221/9/08/p08009. URL: http://dx.doi.org/10.1088/1748-0221/9/08/P08009.

[63] M. Aaboud et al. "Performance of the ATLAS track reconstruction algorithms in dense environments in LHC Run 2". In: *The European Physical Journal C* 77.10 (2017). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-5225-7. URL: http://dx.doi.org/10.1140/epjc/s10052-017-5225-7.

[64] E Abat et al. "The ATLAS Transition Radiation Tracker (TRT) proportional drift tube: design and performance". In: *JINST* 3 (2008), P02013. DOI: 10.1088/1748-0221/3/02/P02013. URL: https://cds.cern.ch/record/1094549.

[65] Morad Aaboud et al. "Study of the material of the ATLAS inner detector for Run 2 of the LHC. Study of the material of the ATLAS inner detector for Run 2 of the LHC". In: *JINST* 12.CERN-EP-2017-081 (2017), P12009. 71 p. DOI: 10.1088/1748-0221/12/12/P12009. URL: https://cds.cern.ch/record/2273894.

[66] "ATLAS liquid argon calorimeter: Technical design report". In: (Dec. 1996).

[67] *ATLAS tile calorimeter: Technical Design Report*. Technical design report. ATLAS. Geneva: CERN, 1996. URL: https://cds.cern.ch/record/331062.

[68] N Ilic. "Performance of the ATLAS Liquid Argon Calorimeter after three years of LHC operation and plans for a future upgrade". In: *Journal of Instrumentation* 9.03 (2014), pp. C03049–C03049. DOI: 10.1088/1748-0221/9/03/c03049. URL: https://doi.org/10.1088/1748-0221/9/03/c03049.

[69] Joao Pequenao. "Computer Generated image of the ATLAS calorimeter". 2008. URL: https://cds.cern.ch/record/1095927.

[70] Thomas Schoerner-Sadenius, ed. *The Large Hadron Collider : Harvest of Run 1*. 2015. DOI: 10.1007/978-3-319-15001-7.

[71] "ATLAS muon spectrometer: Technical design report". In: (June 1997).

[72] A Ruiz Martínez and. "The Run-2 ATLAS Trigger System". In: *Journal of Physics: Conference Series* 762 (2016), p. 012003. DOI: 10.1088/1742-6596/762/1/012003. URL: https://doi.org/10.1088/1742-6596/762/1/012003.

[73] *Technical Design Report for the ATLAS Inner Tracker Pixel Detector*. Tech. rep. Geneva: CERN, 2017. URL: http://cds.cern.ch/record/2285585.

[74] *Technical Design Report for the ATLAS Inner Tracker Strip Detector*. Tech. rep. Geneva: CERN, 2017. URL: http://cds.cern.ch/record/2257755.

[75] *Expected tracking and related performance with the updated ATLAS Inner Tracker layout at the High-Luminosity LHC*. Tech. rep. Geneva: CERN. URL: https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2021-024/.

[76] *Technical Design Report: A High-Granularity Timing Detector for the ATLAS Phase-II Upgrade*. Tech. rep. Geneva: CERN, 2020. URL: https://cds.cern.ch/record/2719855.

[77] G. Avoni et al. "The new LUCID-2 detector for luminosity measurement and monitoring in ATLAS". In: *JINST* 13.07 (2018), P07017. DOI: 10.1088/1748-0221/13/07/P07017.

[78] P Calafiura et al. *ATLAS HL-LHC Computing Conceptual Design Report*. Tech. rep. Geneva: CERN, 2020. URL: https://cds.cern.ch/record/2729668.

[79] *Fast Track Reconstruction for HL-LHC*. Tech. rep. Geneva: CERN, 2019. URL: https://cds.cern.ch/record/2693670.

[80] The ATLAS Collaboration. *ATLAS computing model projections for Phase-II*. Tech. rep. Geneva: CERN. URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults.

[81] ATLAS Collaboration. *Athena*. URL: https://gitlab.cern.ch/atlas/athena.

[82] M Limper. "Track and vertex reconstruction in the ATLAS inner detector". Presented on 12 Oct 2009. 2009. URL: https://cds.cern.ch/record/1202457.

[83] Torbjörn Sjöstrand et al. "An introduction to PYTHIA 8.2". In: *Computer Physics Communications* 191 (2015), 159–177. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2015.01.024. URL: http://dx.doi.org/10.1016/j.cpc.2015.01.024.

[84] Enrico Bothmann et al. "Event generation with Sherpa 2.2". In: *SciPost Physics* 7.3 (2019). ISSN: 2542-4653. DOI: 10.21468/scipostphys.7.3.034. URL: http://dx.doi.org/10.21468/SciPostPhys.7.3.034.

[85] Stefan Höche et al. "QCD matrix elements and parton showers. The NLO case". In: *Journal of High Energy Physics* 2013.4 (2013). DOI: 10.1007/jhep04(2013)027. URL: https://doi.org/10.1007%2Fjhep04%282013%29027.

[86] Andy Buckley et al. "General-purpose event generators for LHC physics". In: *Physics Reports* 504.5 (2011), 145–233. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2011.03.005. URL: http://dx.doi.org/10.1016/j.physrep.2011.03.005.

[87] Stefano Catani et al. "QCD Matrix Elements + Parton Showers". In: *Journal of High Energy Physics* 2001.11 (2001), 063–063. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2001/11/063. URL: http://dx.doi.org/10.1088/1126-6708/2001/11/063.

[88] S. Agostinelli et al. "Geant4—a simulation toolkit". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8. URL: https://www.sciencedirect.com/science/article/pii/S0168900203013688.

[89] Azriel Rosenfeld and John L. Pfaltz. "Sequential Operations in Digital Picture Processing". In: *J. ACM* 13.4 (Oct. 1966), 471–494. ISSN: 0004-5411. DOI: 10.1145/321356.321357. URL: https://doi.org/10.1145/321356.321357.

[90] The ATLAS collaboration. "A neural network clustering algorithm for the ATLAS silicon pixel detector". In: *Journal of Instrumentation* 9.09 (2014), P09009–P09009. ISSN: 1748-0221. DOI: 10.1088/1748-0221/9/09/p09009. URL: http://dx.doi.org/10.1088/1748-0221/9/09/P09009.

[91] *Performance of the ATLAS Silicon Pattern Recognition Algorithm in Data and Simulation at $\sqrt{s} = 7$ TeV*. Tech. rep. Geneva: CERN, 2010. URL: https://cds.cern.ch/record/1281363.

[92] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. ISSN: 0021-9223. DOI: 10.1115/1.3662552. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf. URL: https://doi.org/10.1115/1.3662552.

[93] R. Frühwirth. "Application of Kalman filtering to track and vertex fitting". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 262.2 (1987), pp. 444–450. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(87)90887-4. URL: https://www.sciencedirect.com/science/article/pii/0168900287908874.

[94] C. Runge. "Ueber die numerische Auflösung von Differentialgleichungen". In: *Math. Ann.* 46 (1895), pp. 167–78.

[95] W. Kutta. "Beitrag zur näherungsweisen Integration totaler Differentialgleichungen". In: *Zeit. Math. Phys.* 46 (1901), pp. 435–53.

[96] E. J. Nyström. "Über Die Praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben". In: *Acta Mathematica* 54.none (1930), pp. 185 –204. DOI: 10.1007/BF02547521. URL: https://doi.org/10.1007/BF02547521.

[97] E Lund et al. "Track parameter propagation through the application of a new adaptive Runge-Kutta-Nyström method in the ATLAS experiment". In: *Journal of Instrumentation* 4.04 (2009), P04001–P04001. DOI: 10.1088/1748-0221/4/04/p04001. URL: https://doi.org/10.1088/1748-0221/4/04/p04001.

[98] E Lund et al. "Transport of covariance matrices in the inhomogeneous magnetic field of the ATLAS experiment by the application of a semi-analytical method". In: *JINST* 4 (2008). Approved for publication by the ATLAS Publication comitee, submitted to JINST., P04016. 14 p. DOI: 10.1088/1748-0221/4/04/P04016. URL: https://cds.cern.ch/record/1114177.

[99] Rainer Mankel. "A concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 395.2 (1997), pp. 169–184. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(97)00705-5. URL: https://www.sciencedirect.com/science/article/pii/S0168900297007055.

[100] P F Akesson et al. *ATLAS Tracking Event Data Model*. Tech. rep. Geneva: CERN, 2006. URL: https://cds.cern.ch/record/973401.

[101] M. Aaboud et al. "Reconstruction of primary vertices at the ATLAS experiment in Run 1 proton–proton collisions at the LHC". In: *The European Physical Journal C* 77.5 (2017). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-4887-5. URL: http://dx.doi.org/10.1140/epjc/s10052-017-4887-5.

[102] S. Boutle et al. "Primary vertex reconstruction at the ATLAS experiment". In: *J. Phys. Conf. Ser.* 898.4 (2017). Ed. by Richard Mount and Craig Tull, p. 042056. DOI: 10.1088/1742-6596/898/4/042056.

[103] *Development of ATLAS Primary Vertex Reconstruction for LHC Run 3*. Tech. rep. ATL-PHYS-PUB-2019-015. Geneva: CERN, 2019. URL: https://cds.cern.ch/record/2670380.

[104] G. Aad et al. "Measurement of Higgs boson production in the diphoton decay channel inppcollisions at center-of-mass energies of 7 and 8 TeV with the ATLAS detector". In: *Physical Review D* 90.11 (2014). ISSN: 1550-2368. DOI: 10.1103/physrevd.90.112015. URL: http://dx.doi.org/10.1103/PhysRevD.90.112015.

[105] P. Billoir and S. Qian. "Fast vertex fitting with a local parametrization of tracks". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 311.1 (1992), pp. 139–150. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(92)90859-3. URL: https://www.sciencedirect.com/science/article/pii/0168900292908593.

[106] Giacinto Piacquadio. "Identification of b-jets and investigation of the discovery potential of a Higgs boson in the $WH \to l\nu b\bar{b}$ channel with the ATLAS experiment". Presented on 11 Jan 2010. 2010. URL: https://cds.cern.ch/record/1243771.

[107] Jorgen D'Hondt et al. *Sensitivity of Robust Vertex Fitting Algorithms*. Tech. rep. CMS-NOTE-2004-002. Geneva: CERN, 2004. URL: http://cds.cern.ch/record/787499.

[108] R Frühwirth and W Waltenberger. *Adaptive Multi-vertex fitting*. Tech. rep. CMS-CR-2004-062. Geneva: CERN, 2004. DOI: 10.5170/CERN-2005-002.280. URL: https://cds.cern.ch/record/803519.

[109] David R. Bickel and Rudolf Frühwirth. "On a fast, robust estimator of the mode: Comparisons to other robust estimators with applications". In: *Computational Statistics Data Analysis* 50.12 (2006), pp. 3500–3530. ISSN: 0167-9473. DOI: https://doi.org/10.1016/j.csda.2005.07.011. URL: https://www.sciencedirect.com/science/article/pii/S0167947305001581.

[110] The ATLAS Collaboration. *ATLAS Primary Vertex Reconstruction with ACTS*. URL: https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/IDTR-2021-002.

[111] G Piacquadio and C Weiser. "A new inclusive secondary vertex algorithm for b-jet tagging in ATLAS". In: *Journal of Physics: Conference Series* 119.3 (2008), p. 032032. DOI: 10.1088/1742-6596/119/3/032032. URL: https://doi.org/10.1088/1742-6596/119/3/032032.

[112] *Topological b-hadron decay reconstruction and identification of b-jets with the JetFitter package in the ATLAS experiment at the LHC*. Tech. rep. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2645405.

[113] 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). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-7450-8. URL: http://dx.doi.org/10.1140/epjc/s10052-019-7450-8.

[114] *Optimisation and performance studies of the ATLAS b-tagging algorithms for the 2017-18 LHC run*. Tech. rep. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2273281.

[115] *Secondary vertex finding for jet flavour identification with the ATLAS detector*. Tech. rep. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2270366.

[116] *Optimisation and performance studies of the ATLAS b-tagging algorithms for the 2017-18 LHC run*. Tech. rep. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2273281.

[117] *Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment*. Tech. rep. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2255226.

[118] Xiaocong Ai et al. *A Common Tracking Software Project*. 2021. arXiv: 2106.13593 [physics.ins-det].

[119] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. http://eigen.tuxfamily.org. 2010.

[120] *Boost*. https://www.boost.org. (Visited on 02/03/2021).

[121] Sabrina Amrouche et al. "The Tracking Machine Learning Challenge: Accuracy Phase". In: *The Springer Series on Challenges in Machine Learning* (2019), 231–264. ISSN: 2520-1328. DOI: 10.1007/978-3-030-29135-8_9. URL: http://dx.doi.org/10.1007/978-3-030-29135-8_9.

[122] *Software Performance of the ATLAS Track Reconstruction for LHC Run 3*. Tech. rep. Geneva: CERN, 2021. URL: http://cds.cern.ch/record/2766886.

[123] Sarah Campbell. "sPHENIX: The next generation heavy ion detector at RHIC". In: *Journal of Physics: Conference Series* 832 (2017), p. 012012. ISSN: 1742-6596. DOI: 10.1088/1742-6596/832/1/012012. URL: http://dx.doi.org/10.1088/1742-6596/832/1/012012.

[124] Joseph D. Osborn et al. "Implementation of ACTS into sPHENIX Track Reconstruction". In: *Computing and Software for Big Science* 5.1 (2021). ISSN: 2510-2044. DOI: 10.1007/s41781-021-00068-w. URL: http://dx.doi.org/10.1007/s41781-021-00068-w.

[125] Akitaka Ariga et al. "FASER: ForwArd Search ExpeRiment at the LHC". In: (Jan. 2019). arXiv: 1901.04468 [hep-ex].

[126] A. Alexopoulos et al. "Noninvasive LHC transverse beam size measurement using inelastic beam-gas interactions". In: *Phys. Rev. Accel. Beams* 22 (4 2019), p. 042801. DOI: 10.1103/PhysRevAccelBeams.22.042801. URL: https://link.aps.org/doi/10.1103/PhysRevAccelBeams.22.042801.

[127] M. Baak et al. "HistFitter software framework for statistical data analysis". In: *The European Physical Journal C* 75.4 (2015). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-015-3327-7. URL: http://dx.doi.org/10.1140/epjc/s10052-015-3327-7.

[128] Glen Cowan et al. "Asymptotic formulae for likelihood-based tests of new physics". In: *The European Physical Journal C* 71.2 (2011). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-011-1554-0. URL: http://dx.doi.org/10.1140/epjc/s10052-011-1554-0.

[129] A L Read. "Presentation of search results: theCLstechnique". In: *Journal of Physics G: Nuclear and Particle Physics* 28.10 (2002), pp. 2693–2704. DOI: 10.1088/0954-3899/28/10/313. URL: https://doi.org/10.1088/0954-3899/28/10/313.

[130] Particle Data Group et al. "Review of Particle Physics". In: *Progress of Theoretical and Experimental Physics* 2020.8 (Aug. 2020). 083C01. ISSN: 2050-3911. DOI: 10.1093/ptep/ptaa104. eprint: https://academic.oup.com/ptep/article-pdf/2020/8/083C01/34673722/ptaa104.pdf. URL: https://doi.org/10.1093/ptep/ptaa104.

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

[132] J. Alwall et al. "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations". In: *Journal of High Energy Physics* 2014.7 (2014). ISSN: 1029-8479. DOI: 10.1007/jhep07(2014)079. URL: http://dx.doi.org/10.1007/JHEP07(2014)079.

[133] Richard D. Ball et al. "Parton distributions for the LHC run II". In: *Journal of High Energy Physics* 2015.4 (2015). ISSN: 1029-8479. DOI: 10.1007/jhep04(2015)040. URL: http://dx.doi.org/10.1007/JHEP04(2015)040.

[134] *ATLAS Pythia 8 tunes to 7 TeV data*. Tech. rep. Geneva: CERN, 2014. URL: https://cds.cern.ch/record/1966419.

[135] David J. Lange. "The EvtGen particle decay simulation package". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 462.1 (2001). BEAUTY2000, Proceedings of the 7th Int. Conf. on B-Physics at Hadron Machines, pp. 152–155. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(01)00089-4. URL: https://www.sciencedirect.com/science/article/pii/S0168900201000894.

[136] Leif Lönnblad and Stefan Prestel. "Matching tree-level matrix elements with interleaved showers". In: *Journal of High Energy Physics* 2012.3 (2012). ISSN: 1029-8479. DOI: 10.1007/jhep03(2012)019. URL: http://dx.doi.org/10.1007/JHEP03(2012)019.

[137] W. Beenakker et al. "Stop production at hadron colliders". In: *Nuclear Physics B* 515.1 (1998), pp. 3–14. ISSN: 0550-3213. DOI: https://doi.org/10.1016/S0550-3213(98)00014-5. URL: https://www.sciencedirect.com/science/article/pii/S0550321398000145.

[138] WIM BEENAKKER et al. "SQUARK AND GLUINO HADROPRODUCTION". In: *International Journal of Modern Physics A* 26.16 (2011), 2637–2664. ISSN: 1793-656X. DOI: 10.1142/s0217751x11053560. URL: http://dx.doi.org/10.1142/S0217751X11053560.

[139] Wim Beenakker et al. "Supersymmetric top and bottom squark production at hadron colliders". In: *Journal of High Energy Physics* 2010.8 (2010). ISSN: 1029-8479. DOI: 10.1007/jhep08(2010)098. URL: http://dx.doi.org/10.1007/JHEP08(2010)098.

[140] Simone Alioli et al. "A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX". In: *Journal of High Energy Physics* 2010.6 (2010). ISSN: 1029-8479. DOI: 10.1007/jhep06(2010)043. URL: http://dx.doi.org/10.1007/JHEP06(2010)043.

[141] Wolfgang Lukas. "Fast Simulation for ATLAS: Atlfast-II and ISF". In: *Journal of Physics: Conference Series* 396.2 (2012), p. 022031. DOI: 10.1088/1742-6596/396/2/022031. URL: https://doi.org/10.1088/1742-6596/396/2/022031.

[142] Matteo Cacciari et al. "Top-pair production at hadron colliders with next-to-next-to-leading logarithmic soft-gluon resummation". In: *Physics Letters B* 710.4-5 (2012), 612–622. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2012.03.013. URL: http://dx.doi.org/10.1016/j.physletb.2012.03.013.

[143] Michal Czakon and Alexander Mitov. "Top++: A program for the calculation of the top-pair cross-section at hadron colliders". In: *Computer Physics Communications* 185.11 (2014), 2930–2938. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2014.06.021. URL: http://dx.doi.org/10.1016/j.cpc.2014.06.021.

[144] Nikolaos Kidonakis. "Next-to-next-to-leading-order collinear and soft gluon corrections for t-channel single top quark production". In: *Physical Review D* 83.9 (2011). ISSN: 1550-2368. DOI: 10.1103/physrevd.83.091503. URL: http://dx.doi.org/10.1103/PhysRevD.83.091503.

[145] Nikolaos Kidonakis. "Two-loop soft anomalous dimensions for single top quark associated production with aW-orH-". In: *Physical Review D* 82.5 (2010). ISSN: 1550-2368. DOI: 10.1103/physrevd.82.054018. URL: http://dx.doi.org/10.1103/PhysRevD.82.054018.

[146] Nikolaos Kidonakis. "Next-to-next-to-leading logarithm resummation fors-channel single top quark production". In: *Physical Review D* 81.5 (2010). ISSN: 1550-2368. DOI: 10.1103/physrevd.81.054028. URL: http://dx.doi.org/10.1103/PhysRevD.81.054028.

[147] Stefano Catani et al. "Vector Boson Production at Hadron Colliders: A Fully Exclusive QCD Calculation at Next-to-Next-to-Leading Order". In: *Physical Review Letters* 103.8 (2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.103.082001. URL: http://dx.doi.org/10.1103/PhysRevLett.103.082001.

[148] M. Aaboud et al. "Electron reconstruction and identification in the ATLAS experiment using the 2015 and 2016 LHC proton–proton collision data at $\sqrt{s} = 13$ TeV". In: *The European Physical Journal C* 79.8 (2019). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-7140-6. URL: http://dx.doi.org/10.1140/epjc/s10052-019-7140-6.

[149] *Improved electron reconstruction in ATLAS using the Gaussian Sum Filter-based model for bremsstrahlung*. Tech. rep. Geneva: CERN, 2012. URL: https://cds.cern.ch/record/1449796.

[150] M. Aaboud et al. "Electron and photon energy calibration with the ATLAS detector using 2015–2016 LHC proton-proton collision data". In: *Journal of Instrumentation* 14.03 (2019), P03017–P03017. ISSN: 1748-0221. DOI: 10.1088/1748-0221/14/03/p03017. URL: http://dx.doi.org/10.1088/1748-0221/14/03/P03017.

[151] 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). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-016-4120-y. URL: http://dx.doi.org/10.1140/epjc/s10052-016-4120-y.

[152] Aliaksei Hrynevich. *ATLAS jet and missing energy reconstruction, calibration and performance in LHC Run-2*. Tech. rep. Geneva: CERN, 2017. DOI: 10.1088/1748-0221/12/06/C06038. URL: https://cds.cern.ch/record/2263777.

[153] W Lampl et al. *Calorimeter Clustering Algorithms: Description and Performance*. Tech. rep. Geneva: CERN, 2008. URL: https://cds.cern.ch/record/1099735.

[154] G. Aad et al. "Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1". In: *The European Physical Journal C* 77.7 (2017). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-5004-5. URL: http://dx.doi.org/10.1140/epjc/s10052-017-5004-5.

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

[156] Matteo Cacciari, Gavin P Salam, and Gregory Soyez. "The anti-ktjet clustering algorithm". In: *Journal of High Energy Physics* 2008.04 (2008), 063–063. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2008/04/063. URL: http://dx.doi.org/10.1088/1126-6708/2008/04/063.

[157] M. Aaboud et al. "Jet energy scale measurements and their systematic uncertainties in 13 TeV proton–proton collisions with the ATLAS detector". In: *Physical Review D* 96.7 (2017). ISSN: 2470-0029. DOI: 10.1103/physrevd.96.072002. URL: http://dx.doi.org/10.1103/PhysRevD.96.072002.

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

[159] M. Aaboud et al. "Identification and rejection of pile-up jets at high pseudorapidity with the ATLAS detector". In: *The European Physical Journal C* 77.9 (2017). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-5081-5. URL: http://dx.doi.org/10.1140/epjc/s10052-017-5081-5.

[160] M. Aaboud et al. "In situ calibration of large-radius jet energy and mass in 13 TeV proton–proton collisions with the ATLAS detector". In: *The European Physical*

Journal C 79.2 (2019). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-6632-8. URL: http://dx.doi.org/10.1140/epjc/s10052-019-6632-8.

[161] ATLAS Collaboration. "Expected performance of the 2019 ATLAS b-taggers". In: (2019). URL: http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/FTAG-2019-005.

[162] M. Aaboud et al. "Performance of top-quark and W-boson tagging with ATLAS in Run 2 of the LHC". In: *The European Physical Journal C* 79.5 (2019). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-6847-8. URL: http://dx.doi.org/10.1140/epjc/s10052-019-6847-8.

[163] *Boosted hadronic vector boson and top quark tagging with ATLAS using Run 2 data*. Tech. rep. Geneva: CERN, 2020. URL: https://cds.cern.ch/record/2724149.

[164] M. Aaboud et al. "Performance of missing transverse momentum reconstruction with the ATLAS detector using proton–proton collisions at $\sqrt{s} = 13$ TeV". In: *The European Physical Journal C* 78.11 (2018). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-018-6288-9. URL: http://dx.doi.org/10.1140/epjc/s10052-018-6288-9.

[165] *Object-based missing transverse momentum significance in the ATLAS detector*. Tech. rep. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2630948.

[166] *Simulation-based extrapolation of b-tagging calibrations towards high transverse momenta in the ATLAS experiment*. Tech. rep. Geneva: CERN, 2021. URL: https://cds.cern.ch/record/2753444.

[167] C.G Lester and D.J Summers. "Measuring masses of semi-invisibly decaying particle pairs produced at hadron colliders". In: *Physics Letters B* 463.1 (1999), pp. 99–103. ISSN: 0370-2693. DOI: https://doi.org/10.1016/S0370-2693(99)00945-4. URL: https://www.sciencedirect.com/science/article/pii/S0370269399009454.

[168] Alan Barr, Christopher Lester, and Phil Stephens. "A variable for measuring masses at hadron colliders when missing energy is expected: mT2: the truth behind the glamour". In: *Journal of Physics G: Nuclear and Particle Physics* 29.10 (2003), pp. 2343–2363. DOI: 10.1088/0954-3899/29/10/304. URL: https://doi.org/10.1088/0954-3899/29/10/304.

[169] M. Aaboud et al. "In situ calibration of large-radius jet energy and mass in 13 TeV proton–proton collisions with the ATLAS detector". In: *The European Physical Journal C* 79.2 (2019). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-6632-8. URL: http://dx.doi.org/10.1140/epjc/s10052-019-6632-8.

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

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

[172] William Buttinger. *Using Event Weights to account for differences in Instantaneous Luminosity and Trigger Prescale in Monte Carlo and Data*. Tech. rep. Geneva: CERN, 2015. URL: https://cds.cern.ch/record/2014726.

[173] The ATLAS Collaboration. "Measurements of the production cross-section for a Z boson in association with b-jets in proton−proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector". In: *JHEP* 2007 (2020), 044. 57 p. DOI: 10.1007/JHEP07(2020)044. arXiv: 2003.11960. URL: https://cds.cern.ch/record/2714007.

[174] Sayipjamal Dulat et al. "New parton distribution functions from a global analysis of quantum chromodynamics". In: *Physical Review D* 93.3 (2016). DOI: 10.1103/physrevd.93.033006. URL: https://doi.org/10.1103%2Fphysrevd.93.033006.

[175] Michiel Botje et al. *The PDF4LHC Working Group Interim Recommendations*. 2011. DOI: 10.48550/ARXIV.1101.0538. URL: https://arxiv.org/abs/1101.0538.

[176] Jon Butterworth et al. "PDF4LHC recommendations for LHC Run II". In: *Journal of Physics G: Nuclear and Particle Physics* 43.2 (2016), p. 023001. DOI: 10.1088/0954-3899/43/2/023001. URL: https://doi.org/10.1088/0954-3899/43/2/023001.

[177] Johannes Bellm et al. "Herwig 7.0 / Herwig++ 3.0 Release Note". In: *The European Physical Journal C* 76 (Dec. 2015). DOI: 10.1140/epjc/s10052-016-4018-8.

[178] *Studies on top-quark Monte Carlo modelling for Top2016*. Tech. rep. Geneva: CERN, 2016. URL: https://cds.cern.ch/record/2216168.

[179] Georges Aad et al. "Search for new phenomena with top quark pairs in final states with one lepton, jets, and missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector". In: *JHEP* 04 (2021), p. 174. DOI: 10.1007/JHEP04(2021)174. arXiv: 2012.03799 [hep-ex].

[180] Georges Aad et al. "Search for new phenomena in pp collisions in final states with tau leptons, b-jets, and missing transverse momentum with the ATLAS detector". In: *Phys. Rev. D* 104.11 (2021), p. 112005. DOI: 10.1103/PhysRevD.104.112005. arXiv: 2108.07665 [hep-ex].

[181] *Formulae for Estimating Significance*. Tech. rep. Geneva: CERN, 2020. URL: https://cds.cern.ch/record/2736148.

[182] A. M. Sirunyan et al. "Search for direct top squark pair production in events with one lepton, jets, and missing transverse momentum at 13 TeV with the CMS experiment". In: *Journal of High Energy Physics* 2020.5 (2020). DOI: 10.1007/jhep05(2020)032. URL: https://doi.org/10.1007%2Fjhep05%282020%29032.

[183] Georges Aad et al. "Search for new phenomena in events with two opposite-charge leptons, jets and missing transverse momentum in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector". In: *JHEP* 2104 (2021), 165. 62 p. DOI: 10.1007/JHEP04(2021)165. arXiv: 2102.01444. URL: https://cds.cern.ch/record/2750825.

[184] Alex Sherstinsky. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network". In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306. ISSN: 0167-2789. DOI: 10.1016/j.physd.2019.132306. URL: http://dx.doi.org/10.1016/j.physd.2019.132306.

[185] Serkan Kiranyaz et al. "1D convolutional neural networks and applications: A survey". In: *Mechanical Systems and Signal Processing* 151 (2021), p. 107398. ISSN:

0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2020.107398. URL: https://www.sciencedirect.com/science/article/pii/S0888327020307846.

Acknowledgements

- Aus datenschutzrechtlichen Gründen entfernt -

Curriculum Vitae

- Aus datenschutzrechtlichen Gründen entfernt -