ETP-KA/2023-12

Karlsruher Institut für Technologie

Measurement of Triple-Differential Z+Jet Cross Sections with the CMS Detector at 13 TeV and Modelling of Large-Scale Distributed Computing Systems

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

von der Fakultät für Physik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

DISSERTATION

von

M.Sc. Maximilian Maria Horzela
aus Pforzheim

Tag der mündlichen Prüfung: 24. November 2023

Referent: Prof. Dr. Günter Quast
Korreferent: Prof. Dr. Achim Streit

Contents

CHAPTER 1

Introduction

The origin of particle physics lies in the inherent human curiosity for our surroundings. Since the beginning of recorded history a constant driving question of scientists through all ages was: "What is everything made of?" Prominent examples for theories supposed to answer this question can be found in various cultures and times. Already in Ancient Greece around the fifth and fourth centuries BC the atomic theory was popular among philosophers. The general idea was a physical universe composed out of fundamental invisible atoms in an otherwise empty void, the vacuum. Democritus, the most popular proponent of this theory that survived the millennia, argued that infinitely dividing matter must be impossible and therefore the atom, the smallest indivisible unit, must exist. This argument refusing the existence of infinities in the smallest sounds ad hoc, especially when considering that at the same time he argued that an infinite amount of atoms in infinite types exist. Adversary arguments were made that implied a continuous structure of matter refusing the existence of a void. The most acknowledged authority endorsing a continuous universe was Aristotle. However, at that time the technology was not in a state enabling to study the structure of matter at length scales below the order of millimeters and the dispute between the two theories could not be settled. Consequently, for the rest of the antiquity and the Middle Ages the preferred theory of Aristotelian physics prevailed.

Although, the atomic theory never faded into obscurity and had its short revivals in the alchemical models of the 17th century and further developments in chemistry in the 18th and 19th centuries it remained a theorized thought experiment. Finally, in the 20th century the theory of atoms as discrete constituents of matter could be experimentally verified by Jean Perrin, who tested Albert Einstein's prediction of the Brownian motion originating in the thermal motion of water molecules. Around the same time, Sir Joseph Thompson discovered the electron and proposed an inner structure of the electrically neutral atoms consisting of negatively charged electrons in a positively charged volume. Based on Thompson's findings, Ernest Rutherford formulated the Rutherford model of atoms based on observations in scattering experiments indicating that the positive charge in an atom is concentrated in a small nucleus, containing most of the mass of an atom, orbited by electrons. During the same time, observations of the photoelectric

effect and the black body radiation problem, among other, suggested that the nature of the microscopic constituents of matter do not always obey the physics of classical mechanics. This led to the formulation of quantum mechanics by Max Planck, Albert Einstein, Niels Bohr, Erwin Schrödinger, Werner Heisenberg, Max Born, Paul Dirac and others. Rutherford's atomic model was adapted to be consistent with this quantum theory resulting in the Rutherford-Bohr model.

The observation of nuclear fission – which changes the atoms and creates additional particles in form of radiation – and the creation of multitudes of new types of particles in energetic particle collisions proved that an extension of quantum mechanics was needed. This lead to the formulation of relativistic quantum field theory (QFT). The quantization of fundamental fields in relativistic continuous space-time instead of a fixed set of particles enables to describe the mechanisms for both the relativistic creation and annihilation of particles and their interactions. The Standard Model (SM) of particle physics, a quantum field theory (QFT), describes the fundamental fields, and their interactions via the electromagnetic, strong, and weak nuclear forces with unprecedented precision. The fourth fundamental force describing gravity by the non-quantum theory of general relativity is not included. However, for all conducted measurements so far it has not been proven necessary to include. Nonetheless, it is by far not the end of the road. There are effects related to the theory of particles observed on cosmological scales – for instance the baryon-antibaryon asymmetry observed in the universe – and observations of neutrino oscillations that are not modelled by the SM. Therefore, Beyond Standard Model (BSM) physics are inevitable.

With the start of the Large Hadron Collider (LHC), the most energetic particle accelerator ever built by mankind, the hope was high to discover new particles shedding light on the nature of BSM physics. However, no evidence for BSM physics was found. Instead, the discovery of a new fundamental field predicted by the SM, the Higgs field, further cemented the dominance of the SM as the prevailing theory. All statistically significant measurements favor the SM over models of new physics, so far. As a consequence, the focus of the LHC program switched to the precise measurement of the properties and parameters of the SM. This strategy is fostered by the unprecedented and growing experimental sensitivity of the LHC experiments, and amount of data delivered by the LHC. During this precision phase, the systematic uncertainties related to the methods used for deriving theoretical predictions become dominant. The uncertainties on these methods, however, rely in large parts on precise measurements of SM and empiric parameters that are used as an input to the calculations. Elaborate analyses of independent data recorded by different experiments are conducted, and the resulting observables are compared to precise theoretical predictions to constrain the inputs within the measured precision. Accordingly, the goal is to maximise the precision in the measurement and the corresponding parameter-dependent predictions. Combining measurements in different regions of phase space with complementary sensitivity to parameters can enhance the capability to put constraints. Such an analysis that measures the production cross section of dimuon events in association with jets in a triple-differential phase space at the LHC

is presented in this thesis. The analysed final state can be measured with high precision in the Compact Muon Solenoid (CMS) detector and the measured cross sections will help to constrain theory parameters, for instance parton distribution function (PDF)s, for future analyses.

Both the precision of analysis methods and theoretical calculations correlate with computational expense. Precise analyses require the processing of large amounts of data leading to high demands on storage and computing resources. The derivation of precise theory calculations include computationally expensive algebraic calculations and are often combined with numerical methods. As a consequence, their precision scales with the harnessed processing power and time. To meet the enormous computing requirements of the LHC collaborations, a global distributed computing infrastructure, the Worldwide LHC Computing Grid (WLCG), was founded. It consists of a federation of computing sites connected via a global network. With more precise analyses enabled by increasing amounts of recorded data also the theoretical predictions need to keep up. Consequently, future computing needs for analysis and calculating predictions are expected to increase substantially. This renders efficient usage of computing resources necessary to be able to also meet the future requirements by the LHC collaborations. In a complex and heterogenous infrastructure as the WLCG that is subject to timely changes, however, identifying favorable infrastructure designs is not trivial.

Due to the size of the computing sites in the WLCG building various test infrastructures at production scale just for the matter of direct comparisons are not feasible. Instead, this thesis follows a different approach. By modelling abstractions for the execution of workflows run on the WLCG and implementing these into a simulation tool various execution patterns and infrastructure designs can be tested without the need of building a real world counterpart of each. Thus, theoretical representations of various infrastructure designs can be tested and directly compared based on simulated performance metrics. However, the simulation models need to predict these metrics with sufficient accuracy to be used in real-world applications. This can be tested by validating predictions made with the simulation models against dedicated data measured on real-world systems.

This thesis is structured in two parts. In the first part the analysis of the full dataset recorded by the CMS experiment during the LHC Run 2 at a collision energy of 13 TeV measuring the production cross section of pairs of oppositely charged muons in association with jets in a triple-differential phase space is presented. The underlying theory in form of the SM and the utilized theoretical methods are described in chapter 2. First, the quantum fields and their interactions in the SM are introduced. Next, the perturbative and non-perturbative methods for calculating theoretical predictions are described and the methods for the estimation of related uncertainties are presented. Last, special emphasis is put on one of the empirical parameter sets used as an input for the calculations, the PDF. Chapter 3 describes the LHC and introduces the collider related quantities luminosity and pileup that are important for the measurement of cross sections. In chapter 4, the layers of subdetectors of the CMS experiment, and

the reconstruction of events, contained objects, and related observables are explained. Additionally, the collaborative efforts enabling the vast physics program pursued in the CMS collaboration are acknowledged. Based on this, the analysis of the full LHC Run 2 data collected by the CMS experiment measuring the triple-differential production cross section of dimuon events in association with jets is presented in chapter 5. First, the analysed observables are defined and the reconstruction of these observables is described. Second, the analysed data and the theoretical models used in the interpretation of this data are presented. Third, the procedure for the combination of the data collected at different phases of the CMS detector and the LHC is described and validated. Fourth, the procedure for mitigating detector effects on the analysed differential cross sections is explained and validated. Fifth, the individual and combined statistical and systematic uncertainties related to the various experimental reconstruction methods and the mitigation technique are defined and estimated. Last, the measured cross sections mitigated for detector effects and the assigned uncertainties are compared to two sets of theoretical predictions.

In the second part of this thesis, the modelling of large-scale distributed computing systems is presented in chapter 6. In pursuit to this objective, the distributed computing infrastructure and the computing methods of the WLCG are introduced. Subsequently, methods for the design and performance modelling of such complex distributed infrastructures are discussed. Finally, the modelling of large-scale computing systems is studied. For this purpose, the utilized simulation models and for this work developed models are described. Afterwards, the implementation into a dedicated simulation tool is calibrated and validated, and the computational complexity of the tool is analysed. Last, a study of a large-scale distributed infrastructure design-candidate is performed.

CHAPTER 2

Particle Physics

In the following chapter the theoretical foundations of particle physics and the methods to obtain theoretical predictions are outlined. In section 2.1 the theory of particle physics as a relativistic quantum field theory (QFT) is introduced and the corresponding fields and their interaction are described. In section 2.2 the methods to obtain theoretical predictions for important observables are introduced.

2.1 The Standard Model of Particle Physics

The Standard Model (SM) is a relativistic QFT describing electromagnetism, the weak and strong nuclear forces, the composition of elementary particles building up matter, and their interactions. It describes three of the four fundamental forces with unprecedented accuracy. However, it presents no complete theory of fundamental interactions. For instance, it excludes gravity and fundamentally cannot describe neutrino oscillations. Also, it lacks models for explaining cosmological observations, for example the observed baryon-antibaryon asymmetry in the universe. Nonetheless, no significant observation was found in experiments on earth, except for neutrino oscillations, contradicting the precise predictions made by the SM and pointing to interactions Beyond Standard Model (BSM). In this section a brief description of the SM is given. A more detailed discussion can be found, for instance, in [1] or the standard textbooks, for example in [2–5].

2.1.1 Quantum Fields and Particles

In the SM 17 fundamental quantum fields are included. Excitations to those fields are interpreted as single point-like objects, called *particles*. Their properties follow from the fundamental assumptions posed on the fields. Requiring invariance under Lorentz-transformations leads to a characterization by mass and spin [6]. As a consequence, classes of fermions and bosons with half integer and integer spin, respectively, are obtained from two representations of the Lorentz-group. Fermions ψ obey Fermi-Dirac statistics and bosons ϕ/A_μ Einstein-Bose statistics [7]. In the SM, only spin $\frac{1}{2}$ fermion and spin 0 and 1 boson fields are included.

Additional requirements of invariance of the SM Lagrangian under local transformations, called gauge transformations, lead to further subclassifications of the fields according to additional charges they carry and subsequent interactions they undergo. The SM gauge group is the product of the three special unitary Lie groups

$$SU(3)_C \times SU(2)_L \times U(1)_Y \tag{2.1}$$

with the subscripts C and Y indicating the conserved charge of the corresponding global symmetry and the subscript L indicating that the $SU(2)$ is applied only on the left chiral part of the fermion fields ψ.

Given these restrictions, the SM Lagrangian can be written as

$$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} + i\bar{\psi}\gamma^{\mu} D_{\mu}\psi + h.c. \tag{2.2}$$

without mass terms. The field tensors are defined as

$$iF_{\mu\nu} = [D_{\mu}, D_{\nu}] \tag{2.3}$$

with the commutator indicated by the square brackets and the gauge covariant derivative being

$$D_{\mu} = \partial_{\mu} + igA_{\mu}T \tag{2.4}$$

consisting of the partial derivative of special relativity ∂_{μ}, couplings g and generators of the gauge group T.

The 17 fields are twelve fermions – six quarks and six leptons arranged in three generations of isospin doublets consisting of up- and down-type quarks and electrically charged leptons and uncharged neutrinos, respectively – four gauge bosons, corresponding to the mediators of the electromagnetic, the weak, and strong forces – the photon, the Z boson and W boson, and the gluon – and the scalar H boson.

2.1.2 Interactions

The Lagrangian encodes the full theoretical dynamics of physical processes. As such, all allowed interactions of and resulting observables for particles can be derived and calculated from the Lagrangian. In the SM the terms in the Lagrangian are postulated based on the structure of the Lorentz-group, in order to preserve Lorentz-invariance and the gauge symmetries defined in eq. (2.1). From these simple assumptions the full theory of the SM can be derived.

2.1.2.1 Quantum Chromodynamics

The strong nuclear force is described by quantum chromodynamics (QCD). It is the QFT encoded into the part of the SM Lagrangian that is invariant under $SU(3)_C$ transformations. The corresponding conserved charge C is called colour charge. The $SU(N)$ has

$n^2 - 1$ degrees of freedom and a representation is defined by the generators T^a obeying the algebraic relation

$$\left[T^a, T^b\right] = f^{abc} T_c \ . \tag{2.5}$$

For the $SU(3)$, $a = 1, 2, \ldots, 8$ generators $T^a = \frac{\lambda^a}{2}$ with λ^a corresponding to the Gell-Mann matrices [8] and structure constants f^{abc}. Consequently, with eight gauge fields G^a_μ, called gluons, and $i = 1, 2, 3$ spinors ψ_i, called quarks, the QCD Lagrangian reads

$$\mathcal{L} = \bar{\psi}_i \left(i\gamma^\mu (D_\mu)_{ij} - m\delta_{ij}\right) \psi_j - \frac{1}{4} G^a_{\mu\nu} G^{\mu\nu}_a \tag{2.6}$$

with

$$(D_\mu)_{ij} = \partial_\mu \delta_{ij} - i g_S (T_a)_{ij} G^a_\mu \ . \tag{2.7}$$

The coupling constant of the strong interaction g_S is often replaced by $\alpha_S = \frac{g_S^2}{4\pi}$, and g_S and the mass m of the quarks are free parameters of the theory.

The field tensor of the $SU(N)$ reads

$$G^a_{\mu\nu} = \partial_\mu G^a_\nu - \partial_\nu G^a_\mu + i g_S f^{abc} G^b_\mu G^c_\nu \ . \tag{2.8}$$

Due to the non-vanishing structure constants f_{abc} coupling terms with three and four gauge fields G^a_μ lead to gluon self-interactions. Particles carrying colour charge and thus obeying the interaction rules of QCD, i.e. quarks and gluons, that are not observed as free particles. Instead, only bound states of quarks and gluons which are colour-neutral are observed. This property of QCD at energy scales below $\sim 1\,\text{GeV}$ is called *confinement*. The bound states of quarks and gluons are called *hadrons*. The effective potential in the non-relativistic limit $V(r)$ between two colour charged particles increases proportional to their distance r. This leads to an increased energy density in the strong field between the two particles rendering it energetically favourable to produce new colour charged particles from the vacuum, thus creating new confined states. This property is strongly favored by hadron mass spectroscopy measurements, supporting a potential $V(r) = \kappa r$ with $\kappa \approx 0.2\,\text{GeV}$, and lattice gauge calculations. On the contrary, at high energies and small distances the interactions between colour charged particles can be approximated as interactions between free particles. This property is called *asymptotic freedom*. These interactions between the asymptotically-free particles can be computed in perturbation theory.

The scale-dependent behaviour of QCD interactions is reflected in the energy scale dependence of the coupling constant α_S subject to renormalization in perturbation theory. At high energies α_S decreases to relatively small values ~ 0.1, whereas for low energy scales it increases rapidly, rendering a perturbative description unreasonable. As a consequence, for QCD interactions the *hard* physics at large energy scales are separated from the modelling of the *soft* physics at small energy scales leading to so-called infrared and collinear (IRC) divergences in the perturbative models.

2.1.2.2 Electroweak Theory

The formulation of the electroweak theory (EW) is driven by experimental observations. Due to the observation of parity violation in beta decays [9] it was hypothesized that the weak force only couples to left chiral fermions and right chiral antifermions in charged currents, respectively. From non-divergent pair productions in weak interactions hints to neutral currents were given. Later neutral currents were observed in the interactions of neutrinos with matter [10]. This was described in a vector-minus-axial-vector (V−A) theory. Further observations of combined charge and parity (CP) violations and flavour-changing charged currents lead to the requirement of at least three generations of quarks interacting via the weak force. Transformations of the gauge group $SU(2)_L$ transform only the left-chiral part of fermion doublets ψ_i. The corresponding conserved charge is the so-called isospin I. The right-chiral parts of the fermions are invariant and carry no weak charge. There exist three degrees of freedom, leading to three gauge bosons W^a_μ, related to the generators $\frac{(\tau)^a_{ij}}{2}$ corresponding to the Pauli matrices τ^a and a gauge coupling g. The structure constants of the $SU(2)$ are non-vanishing, leading to self-interactions of the W^a_μ.

Electromagnetic interactions were the first formulated as a QFT. The so-called quantum electrodynamics (QED) is based on gauge transformations under the symmetry group $U(1)$. The associated gauge field is the photon A_μ and the conserved charge is the electric charge q. It couples to electrically charged fermions.

It was discovered that the electromagnetic and V-A interactions can be unified to a combined theory of electroweak theory [11–13]. By replacing the electromagnetic charge in the $U(1)$ with the hypercharge Y and defining the corresponding gauge field B_μ and a coupling g', the covariant derivatives for the combined $SU(2)_L \times U(1)_Y$ gauge group are defined. They read

$$D_\mu \psi^i_L = \left(\partial_\mu \delta^{ij} - ig \frac{(\tau)^a_{ij}}{2} W^a_\mu - ig' \frac{Y}{2} B_\mu \delta^{ij} \right) \psi^j_L \ , \tag{2.9}$$

$$D_\mu \psi_R = \left(\partial_\mu - ig' \frac{Y}{2} B_\mu \right) \psi_R \ , \tag{2.10}$$

with $i = 1, 2$ and $a = 0, 1, 2$. By a linear transformation of the fields one finds the physical Z boson Z_μ, W bosons $W^\pm_\mu$ and photon A_μ of the electroweak theory, defined as

$$\begin{pmatrix} A_\mu \\ Z_\mu \end{pmatrix} = \begin{pmatrix} \cos\theta_W & \sin\theta_W \\ -\sin\theta_W & \cos\theta_W \end{pmatrix} \begin{pmatrix} B^0_\mu \\ W^0_\mu \end{pmatrix}, \tag{2.11}$$

$$W^\pm_\mu = \frac{1}{\sqrt{2}} \left(W^1_\mu \mp W^2_\mu \right) \tag{2.12}$$

with the Weinberg-angle $\theta_W = \arccos \frac{g}{\sqrt{g^2+g'^2}}$. The isospin and hypercharge quantum numbers are related to the electromagnetic charge as

$$q = I + \frac{Y}{2} \,. \tag{2.13}$$

The observation of a range limitation in the weak force suggests massive gauge bosons. However, the addition of mass terms of the form $m_A^2 A_\mu A^\mu$ to the Lagrangian violate the requirement of local $SU(2)_L$ gauge invariance. To nonetheless incorporate mass terms the Higgs mechanism [14–17] was introduced. For the Higgs mechanism a scalar $SU(2)_L$ doublet

$$\phi = \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix} \tag{2.14}$$

is postulated by which the $SU(2)_L \times U(1)_Y$ gauge symmetry is spontaneously broken down to the electromagnetic $U(1)$. The term

$$\mathcal{L}_\phi = |D_\mu^i \phi_i|^2 + V(\phi^\dagger \phi) = |D_\mu^i \phi_i|^2 - \mu^2 \phi^\dagger \phi + \frac{\lambda}{2} \left(\phi^\dagger \phi\right)^2 \tag{2.15}$$

is added to the Lagrangian. Here, D_μ^i denotes the gauge covariant derivative defined in eq. (2.9), and a mass parameter μ and a dimensionless self-coupling constant λ are defined. From the minimum of the potential term $V(\phi^\dagger \phi)$ with respect to $\phi^\dagger \phi$ given by

$$\langle \phi^\dagger \phi \rangle_0 = \frac{\mu^2}{\lambda} =: \frac{v^2}{2} \tag{2.16}$$

the vacuum expectation value v is defined. Expanding the neutral component of the scalar doublet around v and setting the charged component to zero gives

$$\phi = \begin{pmatrix} 0 \\ v + h \end{pmatrix} \,, \tag{2.17}$$

with a real-valued scalar field h. Mass-terms for the gauge bosons emerge from eq. (2.15)

$$\mathcal{L}_{\text{mass}} = \frac{g^2 v^2}{4} W_\mu^+ W^{\mu -} + \begin{pmatrix} A^\mu, & W^{\mu 0} \end{pmatrix} \frac{v^2}{8} \begin{pmatrix} g^2 & -gg' \\ -gg' & g^2 \end{pmatrix} \begin{pmatrix} A_\mu \\ W_\mu^0 \end{pmatrix} \,. \tag{2.18}$$

By rotating the base according to eq. (2.11) the mass of the photon is $m_\gamma = 0$, the Z boson has $m_Z = \frac{v^2}{4}(g^2 + g'^2)$ and the W bosons $m_W = \frac{g^2 v^2}{4}$. Equation (2.15) also defines the interaction of the Higgs field h with the massive gauge bosons.

To assign mass terms to fermions the Yukawa-coupling term

$$\mathcal{L}_{\text{Yukawa}} = -\lambda_f \bar{\psi}_L^i \phi_i \psi_R - \lambda_f \bar{\psi}_R \phi_i^\dagger \psi_L^i \tag{2.19}$$

with a dimensionless coupling constant λ_f specific to the type of fermion is added to the Lagrangian. After the expansion of the complex scalar ϕ (see eq. (2.17)) eq. (2.19) creates mass terms for the fermions with mass $m_f = \frac{\lambda_f v}{\sqrt{2}}$. This term also defines the coupling of the Higgs field to the massive fermions.

2.2 Perturbation Theory and Monte Carlo Predictions

The common method of calculating theoretical predictions for cross sections of particle collision processes and other observables from the Lagrangian of the SM are perturbative calculations. These calculations rely on a perturbative expansion of the time evolution operator

$$U(t, t_0) = \mathcal{T} \exp\left\{-i \int_{t_0}^{t} \mathcal{H}_I(t')\mathrm{d}t'\right\} \tag{2.20}$$

with the time-ordering operator $\mathcal{T}$ and the Hamiltonian in the interaction picture $\mathcal{H}_I = -\mathcal{L}_I$ governing the amplitude for the transition of an initial state $|i\rangle$ at time t_0 to a final state $|f\rangle$ at time t

$$\mathcal{M} = \langle f | U(t, t_0) | i \rangle \ . \tag{2.21}$$

In perturbative calculations $U(t, t_0)$ is expanded in terms of the coupling constants, such that for a given order in those constants a finite number of algebraic terms can be used to construct $\mathcal{M}$. The orders in $|\mathcal{M}|^2$ are typically enumerated as leading order (LO), next-to-leading order (NLO), next-to-next-to-leading order (NNLO), and so forth. Those calculations including terms of the perturbative series up to a designated order in the couplings are also called fixed-order calculations. The individual terms contributing to $\mathcal{M}$ can be obtained using Feynman rules [18] of the contributing theory encoded in the Lagrangian. With those the so-called Feynman-diagrams can be consistently constructed, each corresponding to a contributing algebraic term in the expansion. Still, obtaining the transition amplitude at a fixed order of perturbations is challenging due to the number of contributing terms and individual terms requiring complex integrations over inner degrees of freedom.

Typically, analytical calculations using the transition amplitude (see eq. (2.21)), also called matrix element (ME), as an input involve further integrations of the phase space of final and initial states. Analytical calculations of that kind proof to be very challenging. Therefore, in many cases a numerical integration using so-called Monte Carlo (MC) techniques is performed.

2.2.1 Pseudo-Random Numbers and Monte-Carlo Methods

The benefit of MC techniques for numerical integration, optimization, and modelling of stochastic processes is their scaling with the number of dimensions and an easy implementation of boundary conditions. The base of each MC technique are (pseudo-)random numbers. In most cases, pseudo-random numbers generated by a random number generator (RNG) are used due to the benefit of fast and deterministic generation of random numbers in a reproducible way. There exist a plethora of RNGs that produce uniformly distributed random numbers. Some commonly used examples are reviewed in [19, 20].

Generation of Random Numbers following a Distribution – Generation of a set of random numbers x following a distribution $f(x)$ with normalization $N = \int_{x_{\min}}^{x_{\max}} f(x)\mathrm{d}x$

within a certain interval $[x_{\min}, x_{\max}]$ is of general interest. Given a uniformly distributed random number $r \in [0, 1]$ the relation between $f(x)$ and r reads

$$\int_{x_{\min}}^{x_i} f(x)\mathrm{d}x = \int_0^r dP = \frac{r}{N} \cdot \int_{x_{\min}}^{x_{\max}} f(x)\mathrm{d}x \tag{2.22}$$

with a single random number x_i, also called an event, distributed according to f and event weight $f(x_i)$. In case the cumulative distribution function $F(x)$ of $f(x)$ and the inverse $F^{-1}(x)$ exist eq. (2.22) can be solved for

$$x_i = F^{-1}\left((F(x_{\max}) - F(x_{\min})) \cdot \frac{r}{N} + F(x_{\min})\right) \ , \tag{2.23}$$

requiring only a random number r as an input.

In cases where F or its inverse are not obtainable the *hit-and-miss* method is utilised. Using an upper estimate $g(x) > f(x)$ with a known cumulative distribution function $G(x)$ and inverse events following the distribution of $g(x)$ are generated. The events are accepted with a probability $\frac{f(y)}{g(y)} > r$ with a new random number r. Consequently,

$$\int_{y_{\min}}^{y_{\max}} \frac{f(y)}{g(y)}\mathrm{d}y = \int_{x_{\min}}^{x_{\max}} \frac{f(x)}{g(x)} g(x)\mathrm{d}x = \int_{x_{\min}}^{x_{\max}} f(x)\mathrm{d}x \ , \tag{2.24}$$

leading to the generation of random events y_i following $f(x)$ with weights $\frac{f(y)}{g(y)}$.

MC Integration – An estimate of an integral $I[f]$ of a function $f(x)$ with any dimensionality in x can be obtained without additional effort from the generated events and corresponding weights as

$$I[f] \approx \frac{1}{n} \sum_{i=1}^{n} f(x_i) \ , \tag{2.25}$$

having an uncertainty estimate due to the limited number of generated events n of

$$E[f] = \sqrt{\frac{\mathrm{Var}(f)}{n-1}} \tag{2.26}$$

with variance $\mathrm{Var}(f)$. The convergence of the integration can therefore be improved by either decreasing the variance by a change of variable $x \to y(x)$ as defined above with $\mathrm{Var}(\frac{f}{g}) < \mathrm{Var}(f)$ or increasing the number of generated events n. The former method is known as *importance sampling*. The convergence is independent of the number of dimensions in x.

2.2.2 Perturbative Methods

For observables on particle collisions the squared amplitude of the transition matrix element has to be computed. In perturbation theory the amplitude and its squared are computed in an perturbative expansion in the coupling constants of the theory. Due to the strong coupling only reaching small values at high energy scales (see section 2.1.2) the convergence of the perturbative expansion breaks for small scales. Therefore, the perturbative physics is separated from the non-perturbative parts.

2.2.2.1 Hard Process

The most common observable in particle physics for a process where an initial state of particles $|i\rangle$ transitions to a final state $|f\rangle$ is the *cross section*

$$\sigma_{|i\rangle\to|f\rangle} = \frac{1}{F}\bar{\sum}\int \mathrm{d}\Phi_n \Theta(\Phi_n)|\mathcal{M}|^2 \tag{2.27}$$

with the flux of incoming particles F, averaged sum over all degrees of freedom not observed $\bar{\sum}$, integration over the n particle phase space Φ_n within the integration volume $\Theta(\Phi_0)$.

However, eq. (2.27) holds only for elemental particle states. In QCD compound states of partons make up the initial and final states. Therefore, for hadron collisions eq. (2.27) needs to be complemented to include the transitions of hadrons to the partons contributing in the partonic interaction. It is assumed that the hard interaction at an asymptotic free scale μ_R can be separated from the low energy scale of QCD confinement below the fragmentation scale μ_F. This theorem is called the *fragmentation theorem*, reviewed in [21]. As a consequence the hadron-hadron cross section is defined as

$$\sigma_{hh\to n} = \sum_i \sum_j \int_0^1 \mathrm{d}x_i \int_0^1 \mathrm{d}x_j f_i(x_i,\mu_\mathrm{F}) f_j(x_j,\mu_\mathrm{F}) \int \mathrm{d}\sigma_{ij\to n}(\mu_\mathrm{R},\mu_\mathrm{F}) \tag{2.28}$$

with the so-called *parton distribution function (PDF)* $f_i(x_i,\mu_\mathrm{F})$ encoding the probabilty distributions of partons i within a hadron h with momentum fraction x_i (see section 2.3), and the *partonic cross section* $\sigma_{ij\to n}$ defined by eq. (2.27) with an initial state given by the partons i and j, and n-particle final state. The energy scales μ_R and μ_F are unphysical and there exists no principle that defines the choices of values. However, in most scattering processes some kind of hard scale Q can be identified and chosen for the two scales for instance the invariant mass of the produced particles or a momentum transfer.

2.2.2.2 Parton Shower

The final-state particles produced in the hard scattering process include a handful of partons. Calculations at higher orders in perturbation theory add additional partons to the final state. Consequently, an inclusive number of additional radiated partons is

expected, only limited by the available phase space, with potentially large effects on the phase space occupation of final-state particles. However, it is not feasible to compute the observables in all orders of QCD due to the involvement of expensive calculations. Instead, the corrections by the higher-order terms are included approximately by *resummation* methods. The corrections manifest as logarithmic terms at different energy scales and their contributions are estimated in orders of these logarithms.

Automated resummation methods for leading-logarithmic (LL) terms are introduced by *parton shower (PS)* algorithms. As a bonus, they transition the scale of the final-state particles from the scale of the hard interaction Q to lower energy scales. A detailed description of PS algorithms is given for instance in [22]. In the Sudakov decomposition [23] corrections to the fixed order ME are described as $1 \to 2$ splittings of an original final-state particle j to a daughter particle i at an evolution energy scale t in the so-called *collinear limit*. These kinds of splittings can be sequentially extended leading to the Sudakov form factor

$$\Delta(t, t_c)_i = \exp\left\{ -\int_{t_c}^{t} \frac{\mathrm{d}q}{q} \int_{\frac{t_c}{t}}^{1-\frac{t_c}{t}} \mathrm{d}z \frac{\alpha_S(t)}{2\pi} \mathcal{P}_{ji}(z) \right\} \tag{2.29}$$

with the cutoff scale t_c accounting for the collinear divergence in the splittings and defining thereby resolvable partons and the Altarelli-Paresi splitting functions $\mathcal{P}_{ji}(z)$ [24]. From eq. (2.29) the probability for at least one splitting at scale t is

$$\Pi(t) = \frac{\mathrm{d}\Delta(t, t_c)}{\mathrm{d}t} \ . \tag{2.30}$$

This can be used for implementing a sequence of stochastic splittings with the probability of each step only dependent on the previous one. Starting at a scale $t = Q^2$ of the hard process the probability for a branching at t' is given by eq. (2.30) with t_c replaced by t' if $t' > t_c$. When this holds a z is sampled from $\mathcal{P}(z)$ defining the kinematics of the $1 \to 2$ splitting. This is repeated for all the daughter partons with new $t = t'$ until $t' < t_c$, at which the evolution of the corresponding branching is terminated.

There exists an ambiguity in the choice of the evolution scale t that also indicates the virtuality of the parton evolution. For example t can be chosen as the opening angle between the parent and radiated particle squared θ^2 that is used for deriving eq. (2.29) in the collinear limit $\theta \to 0$. However, any other variable proportional to θ^2 can be used with equivalent leading collinear logarithmic accuracy but will lead to different extrapolations away from the collinear limit and different subleading terms. The choice of $t = k_T^2$, for instance, results in a so-called p_T-ordered shower.

The divergence of the splittings is not only present in the collinear limit, but also for $z \to 0$. This is called the *soft limit*. In contrast to the collinear divergence, the soft divergence is a general feature in QCD and can be factorized out as an universal factor from the amplitude of a hard process. To preserve the picture of independent evolution

of each parton for soft splittings in a collinear PS one can use the opening angle as the evolution scale. This is the base of a so-called angular-ordered shower.

On partons in the initial state the same PS is applied. However, the fact that each radiated parton in the initial state increases the momentum fraction of the original parton from the hadron needs to be taken into account. For this purpose the Sudakov in eq. (2.29) is adjusted

$$\Delta(t, t_c, x)_i = \exp\left\{ -\int_{t_c}^{t} \frac{\mathrm{d}q}{q} \int_{\frac{t_c}{t}}^{1-\frac{t_c}{t}} \mathrm{d}z \frac{\alpha_S(t)}{2\pi} \mathcal{P}_{ij}(z) \frac{x/z f_j(x/z, t)}{x f_i(x, t)} \right\} \tag{2.31}$$

with the PDFs $f_i(x,t)$ using the solutions to the DGLAP equations (see section 2.3) to guide the backward evolution of the initial-state PS [25, 26]. The emitted partons in the initial-state shower produce their own final-state showers.

There exist several implementations of PSs in general purpose MC generators. Typically, these tools implement more than one algorithm, as is discussed for instance in [27–29].

2.2.2.3 Parton-Shower-Matrix-Element Matching and Merging

Fixed-order ME calculations beyond LO in QCD contain real emissions of partons that are also approximately described by the PSs leading to double counting. However, the resummation corrections of the PS cannot be omitted because of large logarithmic contributions in the IRC limit described by the PS. To keep the precision of both the NLO ME calculation and the PS, so-called *matching* methods are utilised to identify the overlap regions and remove them from the ME calculation before running the PS algorithms.

An observable at NLO accuracy with corresponding operator $\mathcal{O}$ can be written as

$$\langle \mathcal{O} \rangle_{\mathrm{NLO}} = \int \mathrm{d}\Phi_n \left(\mathcal{B}(\Phi_n) + \mathcal{V}(\Phi_n) + \int \mathcal{A}(\Phi_{n+1}) \mathrm{d}\Phi_1 \right) \mathcal{O}(\Phi_n) \tag{2.32}$$

$$+ \int \mathrm{d}\Phi_n \int \mathrm{d}\Phi_1 \left(\mathcal{R}(\Phi_{n+1}) - \mathcal{A}(\Phi_{n+1}) \right) \mathcal{O}(\Phi_{n+1}) \tag{2.33}$$

with the Born contribution $\mathcal{B}(\Phi_n)$ and the virtual correction $\mathcal{V}(\Phi_n)$ in the n-particle phase space Φ_n, and the real correction $\mathcal{R}(\Phi_{n+1})$ in the $n+1$-particle phase space Φ_{n+1}. Since $\mathcal{V}$ and $\mathcal{R}$ include IRC divergences the divergent parts are removed by a subtraction function $\mathcal{A}$ modelling the divergent behaviour of $\mathcal{R}$. By comparison to the PS contribution of the first splitting to the same observable,

$$\langle \mathcal{O} \rangle_{\mathrm{PS}} = \int d\Phi_n \mathcal{B}(\Phi_n) \left(1 - \int_{t_c}^{t} \mathcal{P}(\Phi_1) d\Phi_1 \right) \mathcal{O}(\Phi_n) + \int d\Phi_{n+1} \mathcal{B}(\Phi_n) \mathcal{P}(\Phi_1) \mathcal{O}(\Phi_{n+1}) \ , \tag{2.34}$$

the terms filled by both can be identified. As a consequence, the observable can be adjusted by subtraction of the respective PS terms from eq. (2.32), which yields

$$\begin{aligned}\langle\mathcal{O}\rangle_{\mathrm{NLO-PS}} = \int d\Phi_n \Bigg\{ \mathcal{O}(\Phi_n) \left(\mathcal{B}(\Phi_n) + \bar{\mathcal{V}}(\Phi_n) + \int_{t_c}^{t} \mathcal{B}(\Phi_n)\mathcal{P}(\Phi_1) d\Phi_1 \right) \\ - \int d\Phi_1 \mathcal{O}(\Phi_{n+1}) \mathcal{A}(\Phi_{n+1}) \\ + \int d\Phi_1 \mathcal{O}(\Phi_{n+1}) \left(\mathcal{R}(\Phi_{n+1}) - \mathcal{B}(\Phi_n)\mathcal{P}(\Phi_1) \right) \Bigg\} .\end{aligned} \tag{2.35}$$

Equation (2.35) depends on the type of PS and subtraction method used. By replacing eq. (2.32) with eq. (2.35) in the ME calculation and subsequently applying the PS leads to the desired result.

Widely used matching methods are MC@NLO [30] and POWHEG [31]. In case of POWHEG, $\mathcal{P}(\Phi_1)$ is chosen such that $\mathcal{R}(\Phi_{n+1}) = \mathcal{B}(\Phi_n)\mathcal{P}(\Phi_1)$ preserving the full ME accuracy. This makes the POWHEG ME calculation out-of-the-box usable by any PS algorithm. For MC@NLO, the $\mathcal{P}(\Phi_1)$ and subtraction terms $\mathcal{A}(\Phi_{n+1})$ are chosen based on the PS configuration it is matched to. In general, the $\mathcal{P}(\Phi_1)$ and $\mathcal{A}(\Phi_{n+1})$ also include a scale dependence related to the choice of the overlap region at which the perturbative description of the PS is replaced by the ME.

Merging – The choice of the factorization scale μ_F in the ME calculation can be interpreted as a definition of a region of inclusiveness. Below μ_F the ME calculation is considered inclusive in all multiplicities. In contrast, the PS resolves this inclusiveness in the particle multiplicities and leads to an exclusive final state but at a different scale. The distinction between the inclusive and exclusive parts of the perturbative calculation is defined by a free to choose merging scale. In the phase space regions below the scale the exclusive calculation of the PS is valid and above the ME's is. The idea of *merging* algorithms is to identify these regions of phase space in order to combine multiple ME calculations at different parton multiplicities with PS to extend the accuracy of observables sensitive to exclusive multiplicities with ME calculations that are also valid in hard regions of the phase space.

2.2.3 Non-Perturbative Physics

As the PS evolution approaches the cutoff scale $t_c > \Lambda_\mathrm{qcd}$ the strong coupling α_S has large values and the perturbative description breaks down, reaching the region of confinement. For the asymptotically-free partons resulting from the PS a non-perturbative method needs to be defined for their transition to the observable colour-neutral hadrons. With a similar argument, the confined but broken initial-state hadrons that took part in the hard collision need to be treated. Since one parton was removed from these hadrons to take part in the hard scattering the rest of the hadrons remains intact below the fragmentation scale but can take part in further interactions that have to be described by non-perturbative models.

2.2.3.1 Hadronization

The modelling of the transition from colour charged partons to colour-neutral hadrons is based on so-called *fragmentation* models motivated by empirical observations. It is required that these models obey the conservation laws of the SM, i.e. conserve charges, momenta and are Lorentz-invariant. There are two types of models commonly used in general purpose MC generators, the *Lund string model* [32] and the *cluster hadronization model* [33] implemented in Pythia and Herwig, respectively.

The Lund string model is directly motivated by the linear behaviour of the QCD potential in the non-relativistic limit (see section 2.1.2.1). The linear increase of the energy density between two colour charged particles with growing distance motivates the picture of a QCD string between the particles. With enough energy pumped into the particles the string breaks and forms two new strings each corresponding to a new pair of colour charged particles. These two systems move further apart and create a new string in between. This continues until only pairs of particles with separation Δr in distance and Δt in time exist defining the transverse mass of the pair

$$m_T = \kappa^2 \left((\Delta z)^2 - (\Delta t)^2 \right) \ . \tag{2.36}$$

With this the fragmentation function for the hadron final state is

$$f(z) \propto \frac{1}{z}(1-z)^a \exp\left\{ -\frac{b m_T^2}{z} \right\} \tag{2.37}$$

with z being the fraction of the momentum the hadron takes and two free parameters of the model a and b [34]. This is applied until no energy is left. The last hadrons are formed in order to the already created ones. For low mass strings energy and momentum are shuffled across the event in order to enable the hadron to be on the mass shell. Quarks and antiquarks are attached to one string, while gluons are attached to two.

The cluster hadronization model is based on the preconfinement property of PSs [35]. For a scale of any hard process $t \gg t_c$ a universal mass distribution of colour singlet combinations, formed from the partons at the end of the PS dependent only on t_c and Λ_{QCD}, emerges. High masses are suppressed in this distribution by a power law. By non-perturbative splitting at t_c of gluons into quark-antiquark pairs and pairs of diquarks, clusters of quarks and antiquarks with adjacent colour can be formed that are colour-neutral. These clusters correspond to an early form of mesons with low invariant masses. Clusters with masses too high for hadron formation decay into pairs of clusters distributed isotropically in the rest frame of the mother cluster leading to a limited spread in phase space due to the limited mass spectrum of the original clusters. Based on the quark contents of each cluster hadrons are formed. For light clusters momentum and energy can be shuffled in the event to permit the formation of hadrons from these clusters. Additional colour-reconnection models modifying the adjacency structure of oppositely charged quarks introduce interactions between clusters.

2.2.3.2 Underlying Event

By comparison of the predictions made with MC event generators only modelling the hard interaction and all subsequent steps with measured observables sensitive to the hadronic activity in hadron-hadron collisions an underestimation is observed. This can be corrected by addition of hard collisions in the same event. The origin of these additional interactions lies in the remnants of the colliding hadrons after the original hard interaction. Since the hard collision removes a fraction x of the hadrons energy $1 - x$ is left for further collisions. The additional activity introduced by these collisions is called *underlying event (UE)*.

The additional interactions of the partons in the hadron remnants are called *multiple-parton interaction (MPI)*. Due to the high cross section for strong interactions MPIs are mainly parton-parton scatterings producing partons. Since the produced partons carry colour charge the colour structure of the event is changed significantly compared to the hard scattering only event. This can have a major effect on the phase space occupation of hadrons and the hadron multiplicities. Most MPIs are soft and add to the total amount of energy in the collision products of an event. But, the perturbative cross section for MPI is approximately $\mathrm{d}\sigma \propto \frac{\mathrm{d}p_T^2}{p_T^4}$ leading to a considerable cross section in the hard tail for the production of observable jets. The number of MPIs is regulated by the available energy-momentum phase space and colour screening and saturation effects in the PDF of the hadron remnants for $p_T \to 0$. The latter effects, however, are modelled empirically and need to be carefully tuned. The generated perturbative MPIs are passed to PS algorithms and their resulting colour structure adds to the hadronization (see section 2.2.3.1) of the full event. Colour reconnection models lead to correlations between the hard event and especially between individual MPIs.

Although the modelling of MPI is based mostly on perturbative models, MPIs are considered to be non-perturbative effects in this analysis. This labelling is chosen to distinguish the perturbative physics of a fixed-order calculation that does in general not rely on MC methods and therefore does not generate individual events. Consequently, in such calculations there is no method to include a model of UE contributions in the sense of MPI.

2.2.4 Event Generator Tuning

The non-perturbative models, in particular, include free parameters that are not necessarily related to first principles of a theory. In the best case, these models and their parameters are only motivated by phenomenological implications of the theory, often approximated in some limits. In other cases the models are purely empirical. As such, the predictions made by MC event generators and implicitly their models need to be constantly calibrated to measurements of observables sensitive to these models. This process, which can be quite elaborate due to the high number of free parameters, is called *tuning*.

Examples for model parameters that need to be tuned are the parameters governing the soft physics in the fragmentations in the hadronization models (see section 2.2.3.1) and the regulation of MPI (see section 2.2.3.2), but also the renormalization and factorization scales in the calculation of the hard interaction (see section 2.2.2.1) or the cutoff scale of the PS (see section 2.2.2.2). The parameters for the models of soft physics are tuned to data using mostly automated statistical fitting tools that simultaneously optimize the model parameters to numerous measurements. The most common tool used also for the derivation of the tunes in the CMS collaboration for Herwig [36] and Pythia [37] is Professor [38, 39]. Parameters of the hard interaction, however, are mostly tuned or rather chosen independently by the authors of the corresponding algorithms calculating the observables for certain collision processes.

2.2.5 Theoretical Uncertainties

Theoretical uncertainties are assigned due to the ambiguity of the used models introduced by their parameters. The choice of particular values inevitably generates a bias in the predicted result. Therefore, it is common practice to vary the parameters by constant factors and repeat the calculations with the varied parameter values in order to get an estimation of a systematic uncertainty related to the respective model parameters. Further uncertainty estimates for other model parameters are derived by other methods.

Parameters for which an uncertainty estimate is derived by explicit parameter variations are the renormalization and factorization scales μ_R and μ_F (see section 2.2.2.1), the scale t used in the parton shower evolution (see section 2.2.2.2), and the matching and merging scales separating the soft and hard regimes of phase space (see section 2.2.2.3). The μ_R and μ_F are varied independently by a factor f and $\frac{1}{f}$ and the calculations are repeated for all combinations except the two cases where both scales are maximally varied in the same direction. A common choice for the variation factor is $f = 2$. Based on the envelope including all the variations an uncertainty estimate resulting from the scale dependence of the hard process related to the perturbative expansion is derived. The uncertainty is expected to decrease with the inclusion of higher perturbative orders into the perturbative calculation but can increase locally for certain types of processes and parts of phase space. For PS scale variations the same procedure is applied on the scale t. However, the variation is kept constant for all parton splittings, and, consequently, the uncertainty estimate is obtained by only two variations. This also applies to the matching and merging scales. By variations of the scales, respectively, an uncertainty estimate for the corresponding scale is derived.

The uncertainties on the hadronization and underlying event parameters (see sections 2.2.3.1 and 2.2.3.2) and other parameters not covered by the explicit scale variations are derived using the tunes (see section 2.2.4). The tuning procedure gives an estimate for a confidence interval for all tuned parameters. Based on this covariance tune variations can be derived. By variation of the tune parameters in the full calculation for the prediction of

an observable a respective uncertainty estimate is calculated.

Similarly, an uncertainty due to the limited confidence on the PDFs has to be taken into account. The PDFs are derived in a statistical fit that is assigned a confidence interval (see section 2.3). Based on the confidence on the utilised PDF in the perturbative calculation the calculations are repeated with variations within the confidence interval. From these, the uncertainties on the computed observables are derived.

2.2.6 Detector Simulation

To build proper statistical hypotheses for the statistical analysis of data measured at an experiment also the influences of the detectors on the measured observables need to be estimated. This is done by propagating the generated events obtained from the generation chain presented in sections 2.2.2 to 2.2.4 through a simulation of the detector. In this simulation the interactions of the hadrons, photons, and leptons in an event with the detector material are stochastically modelled by MC methods and the corresponding expected signals produced by the readout electronics are simulated as well. The simulation chain implemented by the CMS collaboration is described in [40].

The tool used in the CMS collaboration to model the interactions with the CMS detector is GEANT4 [41]. At first, to achieve realistic events as encountered at the LHC additional generated events are superimposed with the event of interest to add PU contributions. Next, the simulation of the GEANT4 model of the CMS detector includes all kinds of electromagnetic and hadronic interactions of the particles in the events with the active and dead detector parts and keeps track of the deposited charges and produced particles. Finally, the signals of the detector readout electronics based on the deposited charges and particle interactions with the sensors in the detectors are simulated and digitized.

After these simulation steps, a simulated representation of an event as recorded by the real detector is obtained. On this event the same reconstruction steps can be run as for real data.

2.3 Parton Density Distributions

A parton density distribution or parton distribution function (PDF) encodes the non-perturbative confined structure of a hadron. PDFs are used in fixed-order perturbative calculations (see section 2.2.2.1), parton showers (see section 2.2.2.2), and MPIs (see section 2.2.3.2). They are assumed to be universal for a given type of hadron but cannot be derived from first principles by perturbative means. Since perturbative methods break down at scales of confined states of hadrons (see section 2.1.2.1) PDFs have to be estimated from comparisons of theoretical predictions with measured observables. In practice, they are obtained by statistical fits of perturbative predictions to measured data. There exist multiple fitting methods that differ in the parametrization of the fit

models and the chosen data in the fits.

For instance the PDF can be parametrized by orthogonal sets of polynomials or a set of specific functions. The CTEQ collaboration uses a prametrization for PDFs with a flexible form but constraint by phenomenological arguments to derive PDFs, for instance CT10 [42]. Using this parametrization a likelihood is constructed and fitted to a selected set of measured observables. From the covariance of the fit a corresponding fit uncertainty is derived that is used as the uncertainty assigned to the PDFs. A different approach is pursued by the NNPDF collaboration in the derivation of their PDFs, for example NNPDF 3.1 [43]. Instead of specifically choosing a parametrization artificial neural networks are constructed. Due to the large number of degrees of freedom in a neural network these models do not depend on a specific parametrization. Instead, by training the neural network to MC replicas of the selected set of data with an objective function encoding the goodness of fit to the replica a PDF set that matches the data is gradually obtained in the training process. Many independent trainings of neural networks are performed. Each uses a different MC replica of the data leading to slightly different but statistically homologous PDFs parametrized by the network. Each neural network is called a PDF replica. From the collection of replicas an uncertainty on the PDF is constructed.

The observables used in the PDF fits involve measurements from many experiments in different observables. Those measurements are sensitive to processes at different energy scales. Since the PDFs are scale dependent a global fit of the PDFs at all scales is challenging. However, the value of a PDF at a given scale is not independent from the same PDF's value at a different scale. A transition of a PDF from one scale to another is governed by the Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (DGLAP) equations [24, 44, 45]. With these the PDF can be defined at some fixed scale value, typically the mass of the Z boson is used, and transitioned to the needed scale value for a specific observable.

CHAPTER 3

The Large Hadron Collider

The Large Hadron Collider (LHC), described in detail in [46], is a proton and heavy-ion ring accelerator based at the Organisation européenne pour la recherche nucléaire (CERN) near Geneva, Switzerland. It is built underground into the old Large Electron-Positron Collider (LEP) tunnel, see [47], and consists of two almost 27 km long mostly parallel beam pipes. During opertaion they are filled with *bunches*, small spatially confined packets, of hadrons.

Bunches of protons, containing $\mathcal{O}(10^11)$ protons each, are accelerated by 16 radio-frequency cavities in opposite directions to a record energy of 6.5 TeV, breaking the previous record of 6 TeV also held by the LHC. The in this thesis analysed data are records of collisions during the years 2016 to 2018 at a beam energy of 6 TeV. In the year 2016, a maximum number of 2244 bunches, and in the years 2017 and 2018, 2556 bunches were filled simultaneously into each LHC beam pipe with a bunch spacing of 25 ns resulting in a collision rate of approximately 40 MHz. More than 1200 superconducting dipole magnets generating magnetic flux densities of up to 8.3 T guide the beams of charged hadrons on a circular trajectory. Quadrupole, sextupole, octupole and decapole magnets shape the beams to ensure a tight focus reducing the diameter of the beams to approximately 16 ţm at the four collision points.

Around the collision points four detectors record the collision products, Compact Muon Solenoid (CMS) [48], A Toroidal LHC ApparatuS (ATLAS) [49], Large Hadron Collider beauty (LHCb) [50], and A Large Ion Collider Experiment (ALICE) [51]. ATLAS and CMS are multi-purpose detectors designed for investigating particles of the Standard Model (SM) and search for signatures pointing to physics Beyond Standard Model (BSM). ALICE is designed for studying heavy ion interactions and the quark-gluon plasma. LHCb is a forward spectrometer focusing on the study of boosted processes involving bottom-flavour quarks.

3.1 Luminosity

Expected event yields for a certain type of process depend on the laws of nature governing the interactions colliding particles succumb to, and the collision rate supplied by the accelerator. The expected number of events measurable in a detector is

$$N = \sigma \cdot L \tag{3.1}$$

with the cross section σ (see section 2.2.2) and the *integrated luminosity* L. By implication, the rate of events is

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \sigma \frac{\mathrm{d}L}{\mathrm{d}t} = \sigma \mathcal{L}(t) \tag{3.2}$$

with the *instantaneous luminosity* $\mathcal{L}$ depending on the time t. The integrated luminosity is used as a measure of the data collected by an experiment. The instantaneous luminosity is mainly used to refer to the performance the accelerator can provide. It is defined as

$$\mathcal{L} = f \frac{n_1 n_2}{4\pi a_x a_y} \tag{3.3}$$

with the bunch crossing frequency f, the numbers of particles n_i in bunch $i = 1$ and $i = 2$, and the profiles of the overlap of the colliding bunches a_x and a_y in directions x and y transverse to the beam direction.

For a measurement of cross sections it is crucial to determine the integrated luminosity with high accuracy. For this purpose, several independent methods of measuring the luminosity are utilised in the CMS collaboration. The instantaneous luminosity is estimated from the signal rate in the hadronic forward calorimeter (see section 4.1.3). At the same time several dedicated luminosimeters are installed in the CMS cavern measuring for instance the radiation in the cavern or other observables proportional to the collision rate. The information from these independent sources gathered for a certain time-frame are combined and calibrated using the so-called Van-der-Meer scans [52] to obtain a measurement of the corresponding integrated luminosity.

3.2 Pileup

The colliding bunches of hadrons in the LHC are focused to maximise the occurance of collisions with high energy transfers. The respective reduction of the transverse beam profile in that process leads to an increase in the instantaneous luminosity, see eq. (3.3), corresponding to a higher chance of hadrons to interact in a bunch crossing. This higher chance, however, also leads to a greater number of hadron-hadron-interactions per bunch crossing. Due to the typically small cross sections of interactions relevant for the physics program of the LHC experiments compared to the total cross section for hadron-hadron collisions most of these collisions in a bunch crossing are not of interest for the physics

goals of the collaborations. If an interesting event emerges, determined by passing a trigger (see section 4.2.1), all additional collisions happening in the same crossing create signals in the detector overlapping with the signals of the collision of interest. These additional collisions spoiling the sensitivity of the collision of interest are called *pileup*.

The CMS collaboration implements dedicated pileup mitigation algorithms and techniques in the reconstruction of collision events and their products (see section 4.2). The applied mitigation techniques for the analysed dataset for this work are described in detail in [53]. Aditionally, multiple analysis methods are used to monitor the effects of pileup on analysed observables. In the analysis presented in this thesis (see chapter 5), however, no additional emphasis on top of the recommendations by the CMS collaboration is put on the effects of pileup.

CHAPTER 4

The Compact Muon Solenoid

The Compact Muon Solenoid (CMS) apparatus [48] has been described in many publications in great detail, for instance in [54, 55]. Therefore, in this chapter only the most relevant aspects of the detector and the reconstruction of the data recorded with the CMS apparatus by the CMS collaboration are described. First, an overview of the detector and its constituting subdetectors is given in section 4.1. Second, the methods applied for the reconstruction of collision events and its containing objects are described in section 4.2. At last, an overview of the CMS collaboration is given and the efforts of its members are acknowledged in section 4.3.

4.1 The Detector

The CMS detector is one of two multi-purpose detectors at the Large Hadron Collider (LHC) located at Organisation européenne pour la recherche nucléaire (CERN) near Geneva, Switzerland. Its cylindrical shape is partitioned into a central barrel and two endcaps in the outer regions along the beam pipe. The CMS detector is designed to identify electrons, photons, charged and neutral hadrons, and muons. Its central feature is a superconducting solenoid magnet in the barrel providing a homogenous magnetic flux density of approximately 3.8 T inside its volume. Within the volume of the magnet a silicon pixel and strip tracker, a lead-tungstate electromagnetic calorimeter (ECAL) and a steal, brass, and scintillator hadronic calorimeter (HCAL) are installed. Outside gas-ionization chambers are embedded into a steel yoke. The latter limits the expansion of the strong magnetic field and the former detects minimum ionizing particle (MIP)s penetrating the magnet and inner detector volume. At the endcaps forward calorimeters and muon chambers embedded into steel extend the coverage of the detector.

It has been successfully operated since 2010 with the start of LHC Run 1. Since then multiple components have been upgraded and exchanged. In this work, the state of the detector as it has been in place during LHC Run 2 in the years 2016 to 2018 is described.

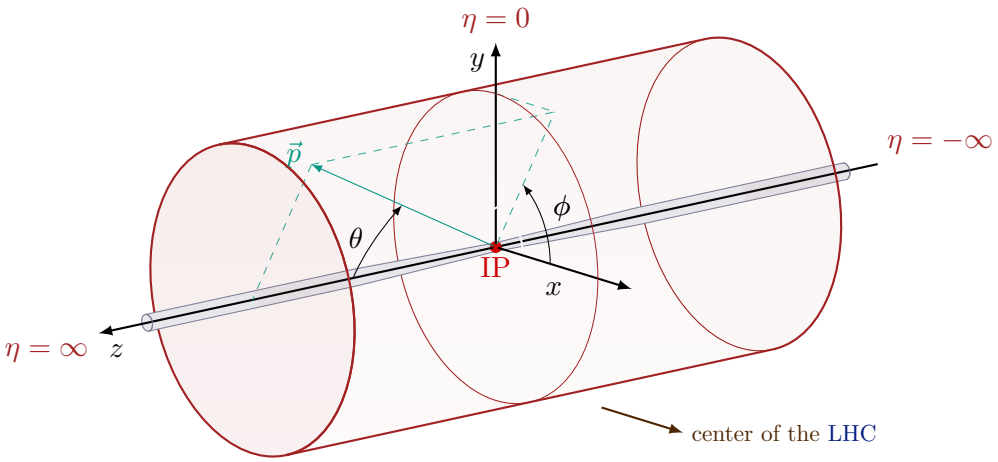


Figure 4.1: Coordinate systems used in the CMS collaboration to describe positions and directions in the detector. The center of the right-handed coordinate system is in the IP. In Cartesian coordinates the x-axis points towards the center of the LHC, the y-axis towards the sky and the z-axis along the beam pipe. In cylindrical coordinates the z-axis is kept the same, while the azimuth angle ϕ in the transverse plane and the polar angle θ are introduced. As an alternative to θ the pseudorapidity η (see eq. (4.2)) can be utilised.

4.1.1 Coordinate System

A right-handed coordinate system with its origin at the interaction point (IP) is used to describe positions and directions in the CMS detector. The coordinate systems used are visualized in fig. 4.1.

In Cartesian coordinates the z-axis is parallel to the beam pipe. This direction is referred to as the *longitudinal* direction. The *transverse* direction is spanned by the x- and y-axes. The x-axis points towards the center of the LHC and the y-axis towards the sky. Due to the cylindrical shape of the detector it is often more convenient to use cylindrical coordinates. For the cylindrical coordinates the z-axis of the Cartesian coordinate system is kept, while the polar angle θ and azimuth angle ϕ are introduced. Consequently a three-vector $\vec{p}$ can be described either in Cartesian or cylindrical coordinates as

$$\vec{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} p \sin\theta \cos\phi \\ p \sin\theta \sin\phi \\ p_z \end{pmatrix} \tag{4.1}$$

with the absolute norm of the vector $p = \sqrt{p_x^2 + p_y^2 + p_z^2}$.

Alternatively, the polar angle θ is substituted by the *pseudorapidity*

$$\eta = -\ln\left(\tan\frac{\theta}{2}\right) \ . \tag{4.2}$$

The pseudorapidity of a particle with energy E and three-momentum $\vec{p}$ is equal to its rapidity

$$y = \ln \sqrt{\frac{E + p_z}{E - p_z}} = \ln \left(\frac{E + p_z}{\sqrt{m^2 + p_T^2}} \right) = \tanh^{-1} \left(\frac{p_Z}{E} \right) \tag{4.3}$$

with the particle's mass m, longitudinal momentum p_z, and transverse momentum

$$p_T = \sqrt{p_x^2 + p_y^2} \tag{4.4}$$

in the limit of $\frac{m}{p} = 0$ which is approximated well for highly relativistic particles as created in LHC collisions.

For two vectors $\vec{p}_1$ and $\vec{p}_2$ with common origin in the IP a metric for their directional distance is commonly used. It is defined in cylindrical coordinates as the distance in the η-ϕ-plane

$$\Delta R = \sqrt{(\eta_1 - \eta_2)^2 + (\phi_1 - \phi_2)^2} \, . \tag{4.5}$$

4.1.2 Tracking

The silicon tracker is the innermost subsystem of the CMS detector closest to the beam pipe and interaction point. It consists of an inner portion consisting of silicon pixel modules and an outer portion consisting of silicon strips.

The pixel detector is exposed to the most radiation. As a consequence, it had to be replaced between 2016 and 2017 data taking. A simultaneous replacement of the beam pipe with a pipe with smaller diameter allowed to add another layer in the innermost part. This additional layer improves the resolution in the measurement of the momenta of particle tracks, compared to the original pixel detector with three layers. A full description of the old pixel detector along with the strip detector is given in [56]. The new design, which has been operational since the data-taking in 2017, consists of four cylindrical barrel layers and three endcap disk layers and is described in detail in [57]. A total of 1856 modules are installed, each equipped with a sensor with 160×416 pixels. The pixels have a size of $100 \times 150\,\mu\text{m}^2$. In total 124 million readout-channels can produce a signal when charged particles traverse through the semiconducter material of the sensor pixels and deposit charge. It covers a phase space up to $\eta = 2.5$ and has a spatial resolution of approximately 9.5 µm and 22.2 µm in the transverse and longitudinal directions, respectively.

The strip detector surrounds the pixel detector and adds another ten layers of silicon sensors in the barrel region and twelve layers in forward and backward directions each. Several layers are enveloped with two layers of strips tilted by a small angle relative to the other facilitating the precise measurement of a three-dimensional hit position. The strip detector has 9.3 million readout channels.

4.1.3 Calorimetry

The calorimeter system surrounds the tracker. In the CMS detector three different technologies are integrated in the calorimetry system to measure the energies of particles. The calorimeter system is split into two parts; the ECAL and HCAL are described in detail in [58] and [59], respectively.

4.1.3.1 Electromagnetic Calorimeter

The ECAL consists of a so-called preshower detector in the innermost layer in the endcaps and lead-tungstate crystals in the barrel and outer endcaps. It is used for measuring the energy of electrons, positrons and photons which deposit their energy in the dense absorber materials via bremsstrahlung. The penetration depth for a material at which an electron deposits a fraction of $\frac{1}{e}$ of its energy is defined as its *radiation length* X_0.

The preshower consists of two layers of lead as absorber material with a total thickness of $3X_0$ and two layers of silicon strip sensors behind the absorbers as a signal layer. They are used for increasing the spatial granularity in the forward region which helps to distinguish between the signatures of two photons produced in π^0 decays and single prompt photons. The preshower detector covers a region in pseudorapidity of $1.653 < |\eta| < 2.6$.

The rest of the ECAL consists of 61200 lead-tungstate ($PbWO_4$) crystals in the barrel and 14648 crystals in the endcaps. The crystals cover a region of $|\eta| < 1.479$ in the barrel and $1.479 < |\eta| < 3.0$ in the endcaps. Due to the placement of readout electronics the coverage is non-continuous for $1.479 < |\eta| < 1.56$. Lead-tungstate is a homogenous calorimeter material simultaneously able to function as a absorber and scintillator. The crystal length corresponds to $26X_0$ and $25X_0$ in the barrel and in the endcaps, respectively. The crystals front facing inside are $2.2\,\text{cm} \times 2.2\,\text{cm}$ in the barrel and $2.8\,\text{cm} \times 2.8\,\text{cm}$ in the endcaps matching approximately the Moliere radius of lead-tungstate such that a photon deposits approximately 94% of its energy in transverse direction in a 3×3 matrix of crystals.

4.1.3.2 Hadronic Calorimeter

The HCAL surrounds the ECAL. It consists of alternating layers of brass or steel absorbers and plastic scintillator tiles or scintillating quartz fibers. It is used for complementing the measurement of the energy of hadrons. Its dimensions are given in units of the nuclear interaction length λ_I defined as the mean distance a hadron can propagate through the material without an inelastic nuclear interaction of the hadron with the nuclei of the material.

In the barrel the absorber material is brass with a total of 40000 plastic scintillator tiles in between. The material budget corresponds to $5.82\lambda_I$ for $\eta = 0$ and $10.6\lambda_I$ for $|\eta| = 1.3$. Just outside the cryostat of the magnet an additional outer calorimeter adds

approximately $3\lambda_I$. The calorimeter in the endcaps is similar to the barrel with slightly smaller dimensions of the scintillators and absorber plates. It extends the phasespace coverage to $|\eta| < 3.0$. Additional forward and backward calorimeters in both directions along the beampipe are located 11 m from the IP and cover the region $2.9 < |\eta| < 5.0$ with absorbers of steel and scintillating quartz fibers.

4.1.4 Muon System

Three different types of gaseous muon detectors are embedded into the steel return yoke outside the magnet and make up the last layer of particle detectors in CMS. The muon chambers are designed to detect MIPs, i.e. muons, which traverse the calorimetry systems and magnet depositing only small fractions of their energy. The muon system is described in detail in [60, 61].

In the barrel region covering $|\eta| < 1.2$, two or three layers of parallel drift tube (DT)s are stacked in orthogonal configuration building up a so-called drift chamber which is able to measure a three-dimensional muon track. Each layer consists of 90 aluminum tubes stacked in four layers with a rectangular profile and an anode wire in the middle. Electrons created in the tube by a muon ionizing the gas inside the tube drift towards the anode wire. Based on the drift time a precise position of the ionization's point of origin can be determined. In combination, a whole muon chamber reaches a spatial resolution of approximately 100 ţm. The time resolution is 5 ns.

In the endcaps where the magnetic field is high and uneven and the muon rate is expected to be larger cathode strip chamber (CSC)s are installed. They consist of six layers of cathode copper-strips crossed with closely spaced anode wires within a gas volume. The anode wires register the drift of electrons while the cathodes do the same for ions created in the ionization of the gas by a traversing muon. The CSCs cover a region of $0.9 < |\eta| < 2.4$. Their spatial and time resolutions are approximately 75 ţm and 6 ns.

In both barrel and endcaps complementary resistive plate chamber (RPC)s are installed covering a region $|\eta| < 2.1$. They consist of two layers of cells built up of separated cathode and anode plates made of an isolator material with gas inbetween. When a muon ionizes the gas electrons drift to the anode creating a signal picked up by metallic strips on the outside of the anode plate. The spatial resolution of an RPC is given by the geometry of the cells and the time resolution is 3 ns.

4.2 Object and Event Reconstruction

Up to three but at least one type of reconstruction workflows is run to convert the electrical signals created in the readout electronics originating from particles interacting with the detector for each collision event to physics objects. The first type involves a crude online reconstruction in sync with the high frequencies of incoming collision data where, based on the properties of the reconstructed objects, a decision on the recording

of the respective events is made. This so-called *trigger* is presented in section 4.2.1. In a subsequent step an in-depth offline reconstruction of the recorded signals to individual physics objects is executed on the distributed computing infrastructure of the Worldwide LHC Computing Grid (WLCG) (see section 6.1.1.1). The reconstruction of these objects is explained in sections 4.2.2 to 4.2.6 for the types of objects relevant for the analysis presented in chapter 5.

4.2.1 Trigger

The first type of reconstructions are run on the intermediate outputs of the detector elements in sync with the incoming collision data. The reconstruction methods present a fast but crude approximation of the full reconstruction. Based on its outcome the respective events are filtered for a final recording of the signals and subsequent full reconstruction or vetoed. The process of keeping the event is colloquially called triggering an event. It is necessary since the full rate of collision events produced at the LHC of 40 MHz cannot be sustained by the data acquisition (DAQ) systems. Therefore, the event rate needs to be reduced to manageable rates of $\mathcal{O}(1\,\text{kHz})$. The trigger system deployed as part of the CMS detectors consists of two stages described in detail in [62].

L1 Trigger – In the level 1 trigger (L1) [63, 64] fast electronic logic devices integrated into the CMS detector perform simple but fast reconstruction algorithms. These algorithms reconstruct electrons and positrons, photons, jets and hadronically decaying tau-lepton candidates from energy deposits in the calorimeters (see section 4.1.3) and muons from signals in the muon detectors (see section 4.1.4). The information of the individual reconstructed objects are combined to create a trigger decision. Each individual algorithm output can be *prescale*d to reduce the corresponding trigger rate emanating from the respective type of reconstructed object. When the event is triggered the detector signals are read-out and transferred to the second trigger stage. The L1 reduces the event rate to approximately 100 kHz.

High Level Trigger – The second trigger stage is the so-called high level trigger (HLT) [65]. It is executed on a computer cluster next to the CMS cavern. The electronic readout signals of the events triggered in the L1 are transferred via high-speed links to individual partitions of the cluster that run simplified versions of the offline reconstruction algorithms optimized for fast processing. For instance, the track reconstruction is only seeded from hits in the pixel detector (see sections 4.1.2 and 4.2.2) and the reconstruction is only performed in certain regions of the detector where L1 primitives suggest a physics candidate. Within these restrictions, the algorithms reconstruct physics objects and directly apply selection criteria on the objects' reconstructed kinematic properties. The HLT reduces the event rate down to approximately 1 kHz. Events that are triggered by at least one HLT algorithm are passed on for permanently storing and full reconstruction of the events.

Single Muon Trigger – One example trigger implemented in the CMS data acquisition workflow [65] is used in the analysis presented in chapter 5. It is the single muon trigger described in [66]. At L1 level the hit information from all muon detectors (see section 4.1.4) are combined to reconstruct muon tracks on dedicated FPGAs. Hits are grouped according to their θ and ϕ coordinates to form tracks. Based on their angular deflexion a crude estimate of the muon's transverse momentum is assigned. After the removal of overlaps the trigger decision is made based on the muon candidate's p_T. The triggered events together with corresponding L1 muon candidates are transferred to the HLT. For muons the HLT is split into two separate steps due to the expensive operation of track reconstruction (see section 4.2.2). In a first step muon tracks are reconstructed with information from the muon systems only, repeating the steps in the derivation of the L1 with a more refined track fit. The obtained tracks are used as a seed for reconstructing the full trajectories of the muons by combining the hits in the muon systems (see section 4.1.4) with the hits in the tracker system (see section 4.1.2). However, due to the limited computing time per event only hits from a small region in the detector, identified based on the L1 and seed information, are taken into account in the fits of the trajectories. In a final step, muon identification and isolation criteria are applied on the resulting muon tracks rejecting non-prompt muons and reducing misidentifications.

4.2.2 Track and Vertex Reconstruction

Charged particles' trajectories are in general reconstructed from hits in the CMS tracker system (see section 4.1.2). A hit corresponds to a deposition of energy by a particle large enough to create a signal in the corresponding sensor cell above the threshold of the readout-electronics.

In the CMS collaboration, a combinatorial algorithm, described in [67, 68] is used. First, based on the iterative algorithm Kalman Filter [69], track candidates are identified from the hits assuming helical paths within the volume of the solenoid starting from a track seed. A seed consists of at least two hits and sets the starting parameters of the track finding algorithm. The Kalman Filter adds hits to the track candidate, first from inside to the outside of the detector and a second time to improve the coherence the other way round using the track candidate as a seed, until no more hits can be found. Second, identification criteria are applied on the track candidates based on the quality of respective fits. For each track candidate a new trajectory is fitted on all selected hits. In this fit effects due to inhomogenousities in the magnetic field and energy losses due to material interactions are taken into account. Hits that are associated with an identified track are removed from the collection of hits in an event reducing the complexity of the next track's reconstruction.

The reconstructed tracks are extrapolated towards the beam axis. Based on their distance to the beam axis a clustering of the tracks based on an annealing algorithm is performed. Each cluster with more than one track is assigned a vertex, a point of generation origin of the corresponding particles and their trajectories are matched to this

vertex. The vertex position is determined by a fit that simultaneously assigns a weight to the tracks encoding the probability for a specific track to be part of its vertex. The primary vertex (PV) is identified as the vertex with the largest sum of transverse momenta of its tracks, as described in [70]. The position of the PV can be determined with a spatial resolution of $< 20\,\mu m$ for a number of tracks > 50.

Electron Track Reconstruction – For the reconstruction of electron and positron tracks the fitting procedure is extended to account for electromagnetic radiation of photons via bremsstrahlung. Those photons can carry a significant portion of the electrons (positrons) energy tangential to the electron trajectory which leads to a significant change in the trajectory's curvature. To account for this effect a different fit procedure is required. It is explained in detail in [71, 72].

4.2.3 Particle Flow

The global event reconstruction in the CMS collaboration is made using the particle flow (PF) algorithm that is described in [72]. It combines the information from all subdetectors to maximise the reconstruction precision. This approach is only possible due to the granularity in the η-ϕ-plane of the individual detector components that allows the identification of single particle candidates. In the PF procedure the objects reconstructed in individual subdetectors, for instance tracks and calorimeter clusters close in the η-ϕ-plane, are linked together to blocks. On each block the same PF algorithm is performed that identifies (groups of) elements in each block that match certain particle criteria. When a match is found the corresponding elements and reconstructed signals are removed from the block and allocated to a reconstructed PF (particle) candidate of the identified type. This is repeated until all elements have been assigned.

There are four types of PF candidates. Muons are identified based on muon tracks (see section 4.2.4). Energy deposits in the calorimeters in vicinity to the muon trajectory with a maximum distance of $\Delta R = 0.3$ are assigned to the respective muon PF candidate. Electrons are identified using an electron track in the tracker system and a matching energy cluster in the ECAL. Additionally, it is required that the energy in linked HCAL clusters does not exceed 10% of the energy deposited in the ECAL. Remaining ECAL clusters that cannot be matched to a track are considered photons. All remaining elements in the blocks are used to create hadrons. Charged hadrons are reconstructed from tracks and matching ECAL and HCAL clusters. All remaining clusters without a matching track are reconstructed as neutral hadrons.

4.2.4 Muons

For muons the tracks obtained from tracker information (see section 4.2.2) are combined with trajectory information from hits in the muon detectors (see section 4.1.4). If an extrapolated particle trajectory reconstructed in the tracker matches to a hit in a DT or CSC it is considered a track of a *tracker muon*. Supplementary, hits in the muon

chambers are fitted to separately reconstruct tracks of *standalone muon*s with the same algorithms as described in section 4.2.2. If the track of the standalone muon can be matched to a track reconstructed from the tracker system the two tracks are combined and refitted. The fitting procedure is described in [73]. This results in a track of a *global muon*.

Tracker muons can originate from particles that were not fully contained in the calorimetry systems and punch through the magnet. Standalone muons can originate from cosmic muons or particle decays in the muon system and contain no vertex information. Therefore, in the analysis presented in chapter 5 only global muons are considered. They combine the precision of the track reconstruction in the tracking system with the precise muon identification and low background in the muon system. This combination of the traits of two detector systems makes global muons the most precisely measurable object with the CMS detector.

The quality of the muon reconstruction is defined by so-called identification and isolation criteria using variables that are sensitive to misidentifications. They are described in detail in [74]. The utilised variables in the identification include the number of hits in the tracker system, the goodness-of-fit of the global muon trajectory, the quality of the match between the inner muon track and the standalone track and the compatibility of the extrapolated tracker track with hits in the muon system. Contributions of particles produced in PU interactions are diminished by applying isolation criteria on the reconstructed muons.

The momentum measurement of the muons is best for high transverse muon momenta $p_T^\mu > 200\,\text{GeV}$ by a combination of the inner track and the track in the muon chambers. For small p_T^μ the momentum resolution is dominated by the momentum resolution of the tracker. Due to the worse performance of the tracker in forward direction resulting in an inaccurate measurement of the curvature of the muon trajectory also the muon momentum resolution decreases. A dedicated calibration of the muon energy scale and resolution is performed to maximise the accuracy of the muon measurement following the descriptions given in [75].

4.2.5 Jets

In the perturbative models used to describe the collision of particles colour charge bearing quarks and gluons (partons) are produced (see sections 2.1 and 2.2). As such they are subject to confinement leading to the formation of showers of color-neutral hadrons in strong relation to the partons they originate from. Possible decays of these hadrons lead to hadrons and leptons hitting the detector in collimated streams. To identify these streams in the CMS collaboration the PF candidates in an event are clustered using a clustering algorithm. These algorithms group the reconstructed PF candidates based on their kinematic properties in order to form observables that correspond to the theoretical picture of a stream of particles originating from a single parton, a so-called *jet*.

4.2.5.1 Jet Clustering

There exist a plethora of jet algorithms which have been used in the era before and during the LHC operation. A nice overview with a discussion of their advantages and disadvantages is given in [76]. In the CMS collaboration the sequential recombination algorithm anti-k_t [77] is used in most analyses. As the name of the class of algorithms imply it sequentially recombines particles to a jet. First, the distances of all particles i to the beam

$$d_{iB} = k_{t,i}^{2p} \tag{4.6}$$

and the distances between particles i and j

$$d_{ij} = \min\left(k_{t,i}^{2p}, k_{t,j}^{2p}\right) \frac{\Delta R_{ij}^2}{R^2} \tag{4.7}$$

with the transverse momentum $k_{t,i}$ of the particle i, the distance in the η-ϕ-plane ΔR_{ij} between particles i and j (see eq. (4.5)), a radius parameter R, and a power factor p are computed. For the anti-k_t algorithm the parameter p is set to -1. Next, the minimum of all d_{iB} and d_{ij} is evaluated. If the minimum is a d_{ij} the two particles i and j are combined into a new particle i. If the minimum is a d_{iB} particle i is called a jet, and it is removed from the collection of particles. This procedure is repeated until all particles are clustered into jets.

Due to the negative sign in p the anti-k_t algorithm favors the clustering of particles with high transverse momenta. Consequently, the jets grow outwards around a hard center leading to a circular shape. Unlike iterative cone algorithms with progressive removal of overlaps, the cones created by anti-k_t are invariant under collinear and infrared radiations of partons. Since collinear radiations are always clustered at the beginning of the sequences the anti-k_t jets present infrared and collinear (IRC) safe observables.

In this work anti-k_t jets with the radius parameter $R = 0.4$ are analysed.

4.2.5.2 Pileup Mitigation

Since the jet clustering takes all PF candidates in an event into account both the particles produced in the proton-proton collision of interest as well as the particles produced in proton-proton collisions in pileup (PU) (see section 3.2) contribute. This effect spoils the interpretation of jet observables related to single hard QCD interactions. The effect of pileup on jets is mitigated in the CMS collaboration in the reconstruction, described in [53].

In this work, the charged hadron subtraction (CHS) technique is applied. It relies on the identification of the PV described in section 4.2.2. All PF candidates whose tracks do not match to the PV are filtered from the event prior to the jet clustering. This removes the contributions of charged particles produced in PU interactions but leaves the neutral

particles. Therefore, further correction procedures are needed to remove remaining PU contributions from the jet observables.

4.2.5.3 Jet Energy Calibration

The PF jets are compound objects combining the information on reconstructed objects measured in all parts of the CMS detector. Although each separate part of the detector and reconstruction chain is carefully calibrated and checked to achieve the best possible accuracy, tiny deficiencies, mismatches and misreconstructions remain. As a consequence, the energy measured in those jets is distorted by the combination of all kinds of tiny, but in their complexity non-negligible, reconstruction and detector deficiencies. Therefore, jets have to be calibrated separately.

The jet energy calibration is performed by the CMS collaboration. The workflow is described in [78, 79]. It consists of a sequence of steps aimed to correct for different sources.

The first step corrects for remaining contributions of PU to the jets. There, the contribution of PU is estimated from the comparison of simulations with and without PU and calibrated with estimates of PU in zero bias data using the *random cone (RC)* method [80].

In a second step the detector response on jets on generator level (see section 5.1.3), jets that are unspoiled by detector and reconstruction effects, is estimated using simulations including the jets at reconstruction level. The jet response, the quotient of the average jet transverse momentum at reconstruction level over the generation level, is derived in bins of η and p_T of the jets corrected for the PU effects.

Lastly, residual corrections accounting for effects not modelled in simulation are derived using dedicated analyses of events with signatures of dijets, multijets, top-antitop pairs, photon plus jets, and oppositely charged muon and electron pairs plus jets. In these analyses, two responses, the direct balance (DB) between a jet and a reference object (ref)

$$R_{\mathrm{DB}} = \frac{p_T^{\mathrm{jet}}}{p_T^{\mathrm{ref}}} \tag{4.8}$$

and the missing transverse energy projection fraction (MPF)

$$R_{\mathrm{MPF}} = 1 + \frac{\vec{E}_T^{\mathrm{miss}} \cdot \vec{p}_T^{\,\mathrm{ref}}}{\left(p_T^{\mathrm{ref}}\right)^2} \tag{4.9}$$

with the missing transverse energy (MET) $\vec{E}_T^{\mathrm{miss}}$ (see section 4.2.6) are utilised. They are derived in bins of η and p_T of the jets corrected with all previous corrections for reconstructed data and simulated events. The information from the different channels

is combined in a fit to derive the final calibration factors for the average of the jet momentum scale.

To match the resolution of the jet momentum in the detector, smearing factors to be applied on simulated jets momenta are derived. They are derived from a measurement of the jet momentum resolution in dijet events. The resolution derived in data is compared to the one in simulation and scale factors are derived.

For the fully calibrated anti-k_t PF jets with radius parameter $R = 0.4$ the momentum resolution is better than 10% and 5% for transverse momenta greater than 100 GeV and 1 TeV, respectively. For the central region of the detector the average jet momentum > 100 GeV can be determined with an uncertainty approximately less than 1%. The uncertainty increases slightly up to tracker coverage $|\eta| < 2.5$. and increases by an order of magnitude in the $|\eta| > 2.5$ regions.

4.2.6 MET

The missing transverse energy (MET) is the negative vectorial sum of all reconstructed PF candidates' p transverse momenta in an event e

$$\vec{E}_T^{\text{miss}} = -\sum_{p \in e} \vec{p}_T^{\,p} \tag{4.10}$$

and corresponds to the transverse momentum of undetected particles. This follows from the momentum conservation. Since the transverse momentum of the initial proton-proton collision can be approximated to be zero, also $\vec{E}_T^{\text{miss}}$ is expected to be zero. However, if particles traverse the detector undetected and therefore no corresponding PF candidate can be reconstructed, a value other than zero is observed. MET is therefore the only observable available at the detector which correlates with the energy of undetectable particles like neutrinos or exotic BSM particles.

However, MET can also originate from detector noise, PU, unshielded activity from outside the detector or misreconstructions. These experimental effects have to be considered in all analyses using MET in their signal model. There are multiple corrections to the MET and event filters based on the MET recommended by the CMS collaboration. One crucial example is given by the corrections applied on jets. In the JEC (see section 4.2.5.3), the jet energies are corrected for PU, detector noise and other experimental and reconstruction effects. These corrections are propagated to the PF-MET $\vec{E}_T^{\text{miss, PF}}$ leading to a corrected MET

$$\vec{E}_T^{\text{miss, corr}} = \vec{E}_T^{\text{miss, PF}} - \sum_{j \in \text{event}} p_T^{j,\text{corr}} - p_T^{j,\text{raw}} \tag{4.11}$$

computed with the corrected and uncorrected transverse momentum $p_T^{j,\text{corr}}$ and $p_T^{j,\text{raw}}$ of all jets j in the event.

4.3 The Collaboration

The CMS collaboration has a broad physics program covering the measurement of SM parameters and interactions, for instance contributing to the discovery of the Higgs boson by the joint effort of the CMS and ATLAS collaborations [81, 82], studies of the complex interactions of heavy-ions, and searches of new interactions and phenomena beyond the Standard Model. With the data collected with the CMS detector since its first operation in 2010 more than 1000 scientific results were collected and published [83]. Reaching this milestone was only possible by the joint efforts of its over 4000 members from 240 institutes in more than 50 countries [84]. In the CMS collaboration physicists, engineers, computer scientists, technicians, and students work together for development and operation of the detector components, the data acquisition, the reconstruction and simulation, and the analysis of the data to conduct cutting-edge research in particle physics.

4.3.1 Computing

A crucial component of the workflows leading to physics results is the management, storage, and processing of the collected data at the CMS experiment. The sheer amounts of data storage and processing requirements of the LHC experiments reaching $\mathcal{O}(100\,\mathrm{PB})$ and $\mathcal{O}(1\,\mathrm{GCPU\text{-}hours})$, respectively [85, 86], are matched by the collaborative, shared, worldwide computing infrastructure of the Worldwide LHC Computing Grid (WLCG). The WLCG enables the scientists to process and analyse the data produced by the experiments. This collaboration of more than 300 sites and its contributions to the successful physics program of the LHC experiments is described in greater detail in section 6.1.

CHAPTER 5

Measurement of Triple-Differential Z+Jet Cross Sections

During the years 2016 to 2018, the Large Hadron Collider (LHC) was operated at a centre-of-mass-energy of 13 TeV that presented record collision energy at the time. This is however, not the only record. The data recorded during that time by the Compact Muon Solenoid (CMS) experiment presents an unprecedented amount available for the discovery of rare collision processes and the precise analysis of already established ones. Such a large dataset enables the precise analysis of processes involving interactions obeying electroweak theory (EW) (see section 2.1) while simultaneously exploring various kinematic regions. These precise differential measurements can be used to compare with precise theory predictions for EW processes, which can be exploited for advancing the understanding of less understood fields of high energy physics (HEP).

The least precisely understood interaction described by the Standard Model (SM) is the strong force described by quantum chromodynamics (QCD) (see section 2.1). While EW interactions can be described by perturbative means with high precision, observables sensitive to interactions of the strong force often rely on empirical models with large uncertainties to be accurately predicted as well. Especially at a hadron collider, where bound states of the fundamental particles of QCD are collided the strong force plays a crucial role. Due to the confinement property of QCD, the precision of the perturbative interpretation of the hard interaction is limited. As a consequence, approximations and empirical models are added for a phenomenological description of processes involving QCD interactions. In order to test and calibrate the phenomenology of the strong force, high-precision measurements of observables impacted by strong interactions need to be studied and compared to the most precise accessible theory predictions.

In this chapter, an analysis of the full dataset of LHC collision events at 13 TeV in the years 2016 to 2018 recorded by the CMS collaboration is presented. In this analysis, the cross sections for the production of oppositely charged muon pairs with an invariant mass close to the mass of a Z boson of 91.1876 GeV [1] in association with jets are measured. Cross sections are measured differentially in three kinematic observables. The measurement strategy is based on the predecessor analyses [87–89] and supersedes their

results. Supplementary, the strategy for a comparison with state-of-the-art perturbative theory predictions for this process is presented. Non-perturbative corrections to these calculations are derived and discussed.

First, in section 5.1 the measured observables, the object- and event-selections, and object- and event-corrections are introduced. Second, in section 5.2 the analysed dataset and the detector conditions, under which the data have been collected, are presented. Third, in section 5.3 the theoretical predictions and simulations utilised for the derivation of the results and their interpretation are discussed. Also, perturbative predictions for the comparison of the measured cross sections and subsequent corrections of these calculations are addressed. Fourth, in section 5.4 the combination of recorded data in the individual data-taking periods to a single analysed dataset is explained after the compatibility of the individual sets is confirmed. The combination procedure for the respective simulations is discussed as well. Fifth, in section 5.5 the procedure for the correction of detector effects is presented and evaluated. Next, in section 5.6.2 systematic effects and subsequent sources for systematic uncertainties are addressed. Finally, in section 5.7 the measured cross sections are presented and compared to theoretical predictions.

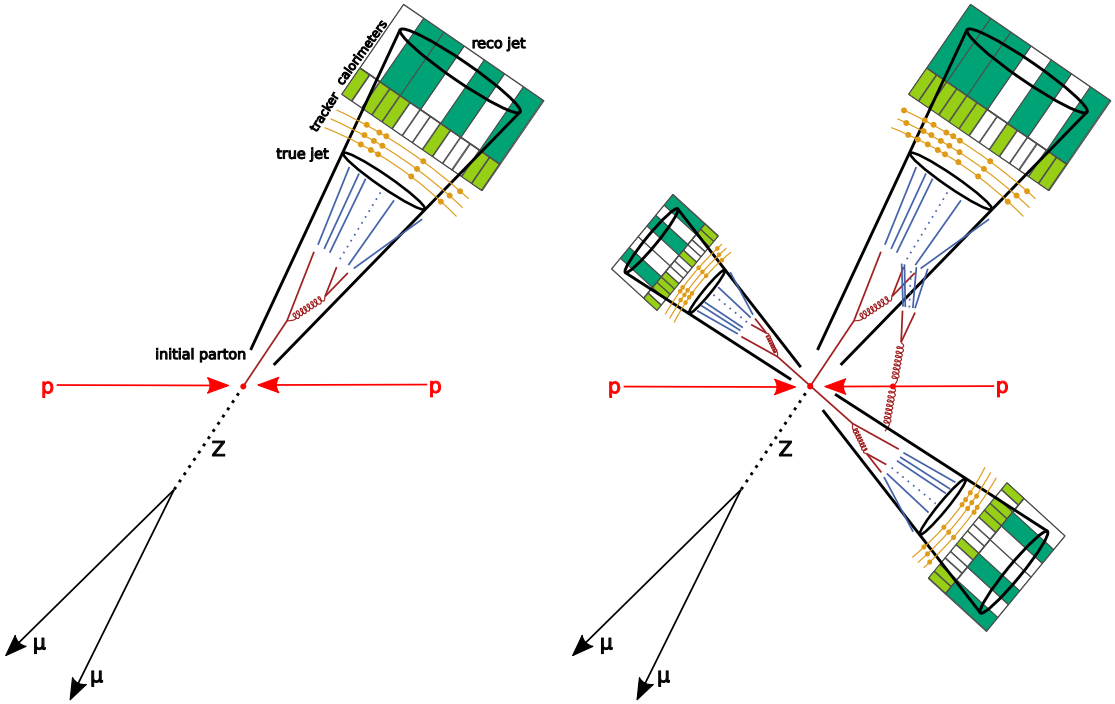
5.1 Analysis Strategy

The goal of the presented analysis is measuring the cross section of the production of oppositely charged muon-pairs with an invariant mass $m_{\mu^+\mu^-}$ within a window of $\pm 20\,\text{GeV}$ around the Z boson mass m_Z^{PDG} [1], in association with at least one jet in proton-proton collisions. The resulting expected topology is sketched in fig. 5.1.

The process is sensitive to the modelling of the strong and electroweak force in the perturbative theories of quantum chromodynamics and electroweak theory. However, the uncertainties in the modelling of QCD are one order of magnitude larger, due to the breakdown of perturbative QCD at small scales. Furthermore, muon pairs in association with jets are frequently generated in collisions at the LHC and therefore pose an important background for searches of new physics and precision measurements in, for example, the Higgs sector. A precise measurement of this process enables further constraints of the parameters of QCD, for example PDFs, and consequently improves the precision of future analyses due to reduced theory-induced uncertainties especially when theory parameters can be overconstrained in a wide kinematic region.

5.1.1 Observables

The cross section is measured differentially in three observables describing the kinematics of the pair of oppositely charged muons, the dimuon system, and the jet with the highest transverse momentum, the hardest jet in the following. Concretely, the cross section is measured in bins of the absolute value of the transverse momentum of the



(a) (b)

Figure 5.1: Sketch of the simplified expected event topology selected in the analysis. The least complex configuration of products of the proton-proton collision (light red arrows) matching the selection criteria (5.1a) includes a pair of oppositely charged muons (black arrows) originating from a Z boson decay, and a jet (black cone). The jet (see section 4.2.5) originates from a single colour charged parton creating a shower of quarks and gluons (dark red straight and looped lines) fragmenting and further decaying into colour-neutral but electrically charged and uncharged hadrons and leptons (blue full and dotted lines). The charged particles leave hits in the tracker system of the detector (orange dots and lines) and deposit their energy in the electromagnetic and hadronic calorimeter system (light and dark green pads). A more frequent and complex configuration (5.1b) involves further hadronic activity in the form of additional jets and pileup. Only the jet with the largest transverse momentum contributes to the analysed observables.

dimuon system p_T^{Z} and the two observables

$$y_b = \frac{1}{2}|y^Z + y^{\mathrm{jet1}}| \tag{5.1}$$

and

$$y^* = \frac{1}{2}|y^Z - y^{\mathrm{jet1}}| \tag{5.2}$$

constructed from the rapidity of the jet with the largest transverse momentum y^{jet1} and the rapidity of the dimuon system y^Z. Since the modelling of the production of oppositely charged pairs of muons at a hadron collider in the analysed kinematic range is dominated by decays of virtual Z bosons the index or superscript Z is used to indicate such dimuon systems. For an ideal configuration of the dimuon system balancing against a single jet a full correlation of y_b with the Lorentz boost and y^* with the scattering angle in the lab frame is given.

By measuring the cross section differentially in an observable correlated with the energy of the collision products, the fundamental parton-parton-interactions in the proton collisions can be probed at multiple scales of the hard interaction. Given that muons are the most precisely measurable object at CMS (see section 4.2.4) the precision of the measurement benefits from an observable that is constructed purely from the muon kinematics, the transverse momentum of the dimuon system p_T^{Z}. Further, dividing the analysed phase space into individual bins of y_b and y^* creates additional sensitivity to the kinematics and flavour contributions of the interacting partons. This sensitivity to the initial-state partons, for instance, enables the imposition of constraints on the PDFs. The predicted contributions of the individual partons' flavours to the analysed phase space bins in p_T^{Z}, y_b and y^* are shown for a fixed-order calculation at NNLO QCD accuracy in [87].

In this thesis, the analysed phase space covers bins of size 0.5 in both y_b and y^* in the ranges $0 < y_b \leq 2.5$ and $0 < y^* \leq 2.5$ with the constraint that the sum of upper bounds of each y_b-y^*-bin does not exceed 3.0. This results in a total number of 15 y_b-y^*-bins. For each of these y_b-y^*-bins one of three binning schemes in p_T^{Z} is assigned. A sketch of the 15 y_b-y^*-bins showing the expected kinematic configuration in the lab frame for the least complex realization with only one jet and the assigned p_T^{Z} binning-scheme is shown in fig. 5.2.

In the idealized topology with only a single jet balancing against the dimuon system, the observables y_b and y^* gain a geometrical interpretation, respectively. At zero y_b and y^* the dimuon system forms a back-to-back topology with the balancing jet with their axis perpendicular to the beam axis. With increasing y_b this event topology is boosted along the beam axis leading to an increased collimation of the dimuon system and the jet. With increasing y^* the angle of the dimuon-jet axis to the beam axis is reduced. Consequently, larger y_b and y^* values lead to higher activity in the forward and backward

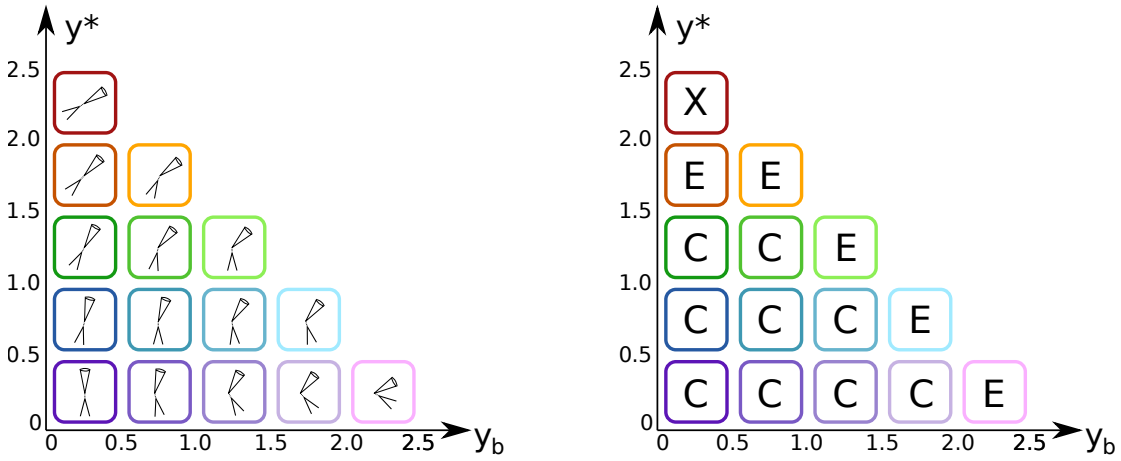


Figure 5.2: Depictions of the kinematic configurations of the idealized dimuon plus jet system (left) and p_T^Z binning schemes (right) for each of the 15 y_b-y^*-bins. The graphics are inspired by [90]. The assigned p_T^Z binning schemes X, E, C correspond to the extreme, edge and central schemes as stated in table 5.1.

detector regions, where the detector efficiency for muons and jets as well as the energy and momentum resolution are reduced compared to the central detector region. This is due to lower precision in the reconstruction of muons and jets in the forward detector (see sections 4.2.4 and 4.2.5) This leads to lower yields and reduced p_T^Z resolution in the outer y_b-y^*-bins.

The binning schemes for p_T^Z in the y_b-y^*-bins have been originally optimized in [87] to ensure large enough event yields in each bin. In bins with high p_T^Z, the limited number of events selected renders it necessary to increase the bin ranges. This effect is more severe for high rapidity bins, where the number of events decreases further making a coarser p_T^Z binning inevitable. In regions of the phase space, for instance for low p_T^Z, where the number of events is not the limiting factor, a minimal bin width is dictated by the limited detector resolution in the corresponding phase space regions. As an ancillary effect, taking the detector resolution into account also suits the unfolding described in section 5.5. The resulting binning schemes are depicted in table 5.1.

In total, the chosen binning results in a total number of 264 measured cross sections. To identify the individual bins, they are indexed starting from the bin with smallest y_b, y^*, and p_T^Z continuing with the next bins in p_T^Z. Once the last p_T^Z -bin is reached, the first p_T^Z-bin in the same y_b-region but with increased y^* is chosen. This is repeated, until the last p_T^Z-bin in the highest y^*-bin is reached and the numbering is continued with the first p_T^Z-bin incremented y_b and lowest y^*. From there, the indexing is continued in the y^* until the last bin is reached again leading to a subsequent increase in y_b. This procedure is continued until all y_b-y^*-p_T^Z-bins are assigned a number between one and 264. This one-dimensional *unravelled* representation of the y_b-y^*-p_T^Z-bins is utilised in

Table 5.1: Three binning schemes for p_T^Z-bins assigned to the y_b-y^*-bins based on the limited detector resolution and statistical precision expected in the corresponding phase space region. In central rapidity regions high numbers of events are expected and the detector resolution is the best. For high rapidities the number of selected events decreases. Therefore, a coarser binning for high p_T^Z is used. In the extreme bin with the highest y^* the smallest number of events is expected. As a consequence, an individual binning scheme for this bin is assigned. The binning follows an optimization made in [87].

binning scheme	p_T^Z bin-edges [GeV]
central (C)	25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 110, 130, 150, 170, 190, 220, 250, 400, 1000
edge (E)	25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 110, 130, 150, 170, 190, 250, 1000
extra (X)	25, 30, 40, 50, 70, 90, 110, 150, 250

the unfolding procedure described in section 5.5.

5.1.2 Event and Object Selections and Corrections

Events in the analysed datasets are filtered by applying selection criteria on the reconstructed objects. The incentive is to obtain a dataset enriched with events containing the prerequisite process signature of two muons and at least one jet in the required regions of phase space. This is done in multiple consecutive steps, each reducing the total amount of data with increasingly more stringent selections. Before the final phase space selection, quality criteria are applied on the reconstructed objects in an event and corrections are applied on the kinematics of the objects to maximise the accuracy of the analysis.

5.1.2.1 Trigger Selection

The first level of selections is performed during the data taking by the trigger system in the CMS detector (see section 4.2.1). However, due to the necessity of a fast trigger decision the trigger relies only on a simplified reconstruction of the events, which renders efficiency corrections necessary that are applied in the subsequent steps.

For this analysis, events are selected that contain at trigger level at least one isolated global muon (see sections 4.2.1 and 4.2.4) above a p_T threshold of 24 GeV for data recorded in 2016 and 2018, or 27 GeV for data recorded in 2017. The p_T threshold for 2017 is increased due to a higher threshold for the unprescaled trigger (see section 4.2.1).

For simulation the same selection on the simulated trigger decision is applied.

Table 5.2: Selection criteria applied on the meta-data of the muon reconstruction for the tight global muon identification as recommended by the CMS collaboration for the analysis of LHC Run 2 data [74].

Muon reconstruction variable	Selection cut
χ^2/ndf of the global-muon track fit	< 10
Number of hits in the muon system included in the global muon fit	≥ 1
Number of hits in the muon system	≥ 2
Number of hits in the pixel tracker	≥ 1
Number of hits in the pixel or strip tracker	≥ 6
Transverse distance of muon track to primary vertex d_{xy}	$< 0.2\,\text{cm}$
Longitudinal distance of muon track to primary vertex d_z	$< 0.5\,\text{cm}$

5.1.2.2 Object Selections

Before applying further selections, quality criteria on the reconstructed particle flow (PF) candidates (see section 4.2.3) contained in the events are applied. They reduce the impact of detector signals not originating from the collision, like noise and malfunctions in the individual detector cells, pileup (PU) (see section 3.2), and *beam-halo*, particles produced by interactions of the beam with the accelerator. Furthermore, they also reduce the contribution of events with incorrectly reconstructed and classified objects. The quality criteria applied in this analysis are presented in the following.

Muons – To reduce the effect of misidentification of PF candidates originating for instance from charged hadrons or electrons misinterpreted as muons, a so-called tight muon identification procedure is applied on global muons (see section 4.2.4). The procedure consists of seven selection criteria recommended by the CMS collaboration for the analysis of the data recorded during LHC Run 2 [74], which are applied on the variables used in the reconstruction of the muon (see section 4.2.4). For a PF muon to be considered as a valid muon in the analysis it is therefore required to pass all the selection criteria. A summary of the applied criteria is given in table 5.2.

To suppress the misidentification of non-prompt muons originating from hadron decays in jets, an isolation criterion is applied on the muon candidates following the CMS recommendations [74]. A selection is applied to the PF isolation [72]

$$I^{\mu}_{\text{PF}} = \frac{1}{p^{\mu}_{T}} \left(\sum_{h^{\pm} \in \text{PV}} p^{h^{\pm}}_{T} + \sum_{\gamma} p^{\gamma}_{T} + \sum_{h^{0}} p^{h^{0}}_{T} - 0.5 \sum_{h^{\pm} \in \text{PU}} p^{h^{\pm}}_{T} \right) \tag{5.3}$$

which is computed from the scalar sum of all transverse momenta of PF candidates within a cone around the muon with

$$\Delta R = \sqrt{(\Delta\phi)^2 + (\Delta\eta)^2} < 0.4 \; . \tag{5.4}$$

Table 5.3: Selection criteria applied on jet constituent variables for the tight jet identification following the recommendations given by the CMS collaboration. The selection cuts are applied on variables derived from the jet constituents, i.e. neutral and charged hadrons, charged leptons (muons and charged electromagnetic) and photons (neutral electromagnetic).

Jet constituent variable	Selection cut
Neutral hadron energy fraction	< 0.9
Neutral electromagnetic energy fraction	< 0.9
Muon energy fraction	< 0.8
Charged hadron energy fraction	> 0
Charged electromagnetic energy fraction	< 0.8
Number of constituents	> 1
Number of charged constituents	> 0

Equation (5.3) takes as inputs the collections of charged hadrons originating from the primary vertex (PV) $h^{\pm} \in \mathrm{PV}$, all neutral hadrons h^0 and photons γ. The contribution of PU is estimated from the charged hadron contribution from PU $h^{\pm} \in \mathrm{PU}$ multiplied by 0.5, as recommended by the CMS collaboration, and subsequently subtracted. The resulting contributions of all particles gets divided by the transverse momentum of the muon p_T^{μ} to obtain the isolation value. The obtained isolation for a muon I_{PF}^{μ} is required to be less than 0.15. With this selection, 95% of all muons originating from hadron decays are vetoed.

Jets – Jets are reconstructed by clustering all charged PF candidates reconstructed from the PV and all neutral PF candidates (see section 4.2.5). This compound nature makes the misidentification of proper jets subject to various sources like detector noise, misreconstruction, and mismodelling.

Reconstructed jets overlapping with problematic regions of the detector which are known to produce anomalous contributions from various sub-detectors in specific η-ϕ-regions of the detector, identified by the CMS collaboration, are considered to be flawed. Therefore, when the hardest jet in an event matches such a η-ϕ-region, the event is vetoed. It is applied on both data and simulation. This removes certain parts of the phase space from the analysis. Consequently, the jet veto has a similar effect on the detector acceptance as the ordinary phase space selections (see section 5.1.2.4) that are mitigated in the unfolding method presented in section 5.5.

To suppress the identification of jets originating from detector noise, miscalibration of detector components, or misreconstructed objects, quality criteria based on the jet composition are applied. These identification criteria take the numbers of different types of PF candidates, i.e. charged and neutral hadrons, charged leptons and photons, and their share of the total jet energy into account. The applied selection cuts are listed in table 5.3. According to the recommendations by the CMS collaboration no efficiency

corrections for the jet identification are needed.

To reduce the effect of jets originating from PU spoiling the analysed event topology a boosted decision tree based classification is performed. The so-called *PU jet identification (PUJetID)* uses multiple jet and event variables to classify the origin of the corresponding jet. When the jet's origin is identified to be PU, the jet is not considered in the analysis. It is only applied on jets with an absolute transverse momentum below 50 GeV.

Since jet clustering takes all PF candidates as input into account, also muons are clustered into jets. To avoid double counting jets overlapping with one of the prompt muons within a radius of 0.4 in η-ϕ-space are removed from the collection of jets considered in the analysis.

Missing Transverse Energy – Detector signals induced by particles not originating from the analysed collision can have a measurable effect on the missing transverse energy in an event (see section 4.2.6). Any activity in the detector introduced from other sources than the original proton collision and mistakenly associated to the analysed collision increases the MET. Therefore, the CMS collaboration provides a collection of dedicated algorithms to run on all events in order to identify events with undesired contributions. Such events in measured and simulation are vetoed. The effect on the selected event yields is much smaller than 1%.

5.1.2.3 Energy and Efficiency Corrections

The measured kinematic observables and the reconstruction and selection efficiencies in data and simulation do not exactly agree. This is expected since the simulation is always an approximation to the real world. Therefore, the measured energies and momenta of the analysed objects are calibrated to improve the agreement between data and simulation. Furthermore, the results of (object) selections differ between data and simulation leading to differences in the selected event yields. Therefore, the efficiency in the simulation is corrected to the one in data to reproduce the measured yields in simulation. The total efficiency correction is acquired on an event-by-event base. It is applied by multiplying correction weights for the individual sources to each event weight respectively, following the Poisson statistic of weighted events [91]. Later, the corrected efficiency is treated together with acceptance effects in the unfolding (see section 5.5). In the following, the different corrections are described in detail.

Muon Momentum Calibration – In a first step the momentum of the identified and isolated muons is corrected for detector misalignment, unanticipated magnetic field effects in simulation and bias in the reconstruction utilising the precisely known mass of the Z boson (see section 4.2.4). Correction factors from an independent analysis of dimuon events for each analysed year of data are provided by the CMS collaboration. They depend on the charge and the η-ϕ-region of the detector. Two factors are provided

to adjust on average the muon momentum distribution in data to the value observed in simulation and smear the momentum resolution in simulation to the observed resolution in the detector. The momentum resolution is intentionally overestimated in the simulation since a degrading of the resolution can be easier amended than an enhancement.

The resulting muon candidates are further corrected for electromagnetic radiation of photons. For a corrected muon, the momenta of all photons within a cone with radius $R = 0.1$ around the muon in η-ϕ-space are added to the momentum of the muon. Muons corrected this way are called *dressed* muons.

Jet Energy Calibration – Similar to the muon momentum calibration the energy scale and resolution of the reconstructed jets is calibrated. The jet energy calibration involves several levels due to the composite nature of the jets (see section 4.2.5.3).

The energy scales for jets in simulation are corrected for PU effects and differences in the reconstruction of the jet momenta in data and simulation. The energy scales for jets in data are additionally corrected for residual detector and reconstruction effects not accounted for in simulation. The factors for the correction of the energy scale of the jets' momenta in both data and simulation together with corresponding uncertainty estimates are provided by the CMS collaboration.

The jet resolution in simulation is calibrated to the one in data by scaling the simulated jets' momenta with corrected energy scale. Two cases and corresponding corrections are distinguished depending on whether a jet at reconstruction level can be matched to a jet on generator level. A jet in simulation can be matched to a jet at generator level if the distance between the jets in the η-ϕ-plane at reconstructed reco and generated gen level overlap in η-ϕ-space satisfied by

$$\Delta R < \frac{R}{2}$$

with jet radius parameter of the anti-k_T clustering algorithm $R = 0.4$ and

$$|p_T^{\text{reco}} - p_T^{\text{gen}}| < \sigma_{\text{JER}} p_T^{\text{reco}}$$

with the measured jet resolution in data σ_{JER}. In that case, the simulated jet's momentum is scaled by

$$c_{\text{JER}} = 1 + (s_{\text{JER}} - 1)\,\frac{p_T^{\text{reco}} - p_T^{\text{gen}}}{p_T^{\text{reco}}} \tag{5.5}$$

with the resolution scale factor s_{JER} and corresponding uncertainty estimates provided by the CMS collaboration. Both, s_{JER} and σ_{JER} are evaluated for the η and p_{T} of the jet at reconstruction level. When no jet at generator level can be matched, the momentum is smeared stochastically with

$$c_{\text{JER}} = 1 + r\sqrt{\max\left(s_{\text{JER}}^2 - 1, 0\right)} \tag{5.6}$$

with a random number r drawn from a normal distribution with expectation value 0 and variance σ^2_{JER}.

Muon Efficiency Corrections – Efficiency corrections for the selection of triggered events with an identified isolated muon in data and simulation together with corresponding uncertainty estimates are provided by the CMS collaboration. The efficiency measured in simulation is scaled to the efficiency measured in data.

The efficiencies are derived with the tag-and-probe method [73]. The total muon efficiency is factorized into four components which are derived individually. They include the efficiencies for

- reconstructing a muon track in the tracker and the reconstruction of a muon from this track ϵ_{reco} [67],
- identifying a muon according to the quality criteria ϵ_{ID} (see section 5.1.2.2),
- identifying a muon according to the isolation criteria ϵ_{Iso} (see section 5.1.2.2),
- and the efficiency for triggering the event with the muon trigger ϵ_{trig} (see section 5.1.2.1).

Since the single muon trigger only requires a single muon but two muons (μ^1 and μ^2) are selected at analysis level, the efficiency for the muon trigger is

$$\epsilon_{\text{trig}} = 1 - \left(\prod_{i=1}^{2} 1 - \epsilon_{\text{trig}}(\mu^i) \right) \ . \tag{5.7}$$

The other efficiencies are considered to be fully correlated between the two selected muons. Therefore, the total muon efficiency is given by

$$\epsilon = \epsilon_{\text{trig}} \prod_{i=1}^{2} \epsilon_{\text{reco}}(\mu^i)\epsilon_{\text{ID}}(\mu^i)\epsilon_{\text{Iso}}(\mu^i) \ . \tag{5.8}$$

Prefiring Correction – During data taking in the years 2016 and 2017 a gradual timing shift in the electromagnetic calorimeter (ECAL) level 1 trigger (L1) trigger primitive (TP) was observed for the ECAL region with $\eta > 2.0$. This caused corresponding TPs for $\eta > 2.0$ to be assigned to the previous bunch crossing. Remaining TPs firing in the unaffected region lead to an increased probability of causing the L1 to fire on two consecutive collision events. Since the L1 directives do not allow two consecutive events to be triggered this leads to a veto on the otherwise accepted events resulting in an efficiency loss of triggered events. To mitigate this efficiency loss, the CMS collaboration provides η- and p_T-dependent probabilities $p^i(\eta^i, p^i_T)$ for the prefiring to happen for a particle i. They are derived from unaffected events and are meant for scaling the yields of events with photons and jets in regions affected by prefiring.

A similar issue was also present in the muon L1 during the whole period of data taking of the analysed dataset. Due to the limited time resolution of the muon systems (see section 4.1) a non-vanishing probability for the assignment of a muon candidate to the wrong bunch crossing existed. This led to vetoing of events due to the L1 directives. This effect is more severe in the data taken in 2016 but is non-negligible in the other years as well and affects the whole phase space covered by the muon system. For this additional prefiring effect, the CMS collaboration provides probabilities for certain classes of objects subject to prefiring effects together with corresponding uncertainty estimates as well.

From the provided probabilities, an event weight based on its particle composition is derived in simulation. The total probability of prefiring not to happen is given by

$$w = \prod_{i \in (\gamma, \text{jets}, \mu)} 1 - p^i(\eta^i, p_T^i) \tag{5.9}$$

with i being a reconstructed object from the collection of photons γ, jets, or muons μ in the corresponding event. It is multiplied to each event's weight subject to prefiring to reweight the distributions in simulation to the ones observed in data.

PUJetID Efficiency Correction – Differences in the efficiencies of the identification of PU jets in simulation and data are corrected by reweighting the events in simulation. The event weights are given by

$$w = \prod_{i \in \text{matched jets}} p^i \left(\eta^i, p_T^i\right) \tag{5.10}$$

with η- and p_T-dependent efficiency scale-factors p^i for each jet i in an event subject to the PUJetID and matched to a corresponding jet on generator level assuming full correlation between the individual jets. The scale factors p^i together with corresponding uncertainty estimates are provided by the CMS collaboration.

Other Efficiency Corrections – Following the recommendations by the CMS collaboration, efficiency corrections for the jet identification criteria described in section 5.1.2.2 are omitted.

Due to the negligible effect of the MET filters on the event selection (see section 5.1.2.2), efficiency corrections for the MET filters are neglected.

5.1.2.4 Phase Space Selections

Final analysis selections are applied on the quality-assessed and corrected objects to enrich the analysed data with events matching the signal criteria and phase space and to suppress background events. The selections are tighter than the ones imposed by the trigger to avoid a bias introduced by the trigger selection based on crude reconstruction.

Table 5.4: Overview of all final selection criteria (after all quality criteria and corrections) applied to muons and systems of oppositely charged muon pairs in the analysed events. Each criteria is applied on the respective identified and corrected collections of reconstructed muons and pairs of muons.

Quantity	Selection Criteria
p_T^{μ}	$> 29\,\text{GeV}$
$\lvert\eta_{\mu}\rvert$	< 2.4
$N_{\mu^+\mu^-}$	≥ 1
$m_{\mu^+\mu^-}$	$> 71.1876\,\text{GeV} \wedge\ < 111.1876\,\text{GeV}$
$p_T^{\mu^+\mu^-}$	$> 25\,\text{GeV}$

They are applied on the fully corrected objects passing all previous selection steps and are run after all quality criteria and corrections have been deployed. The resulting event yields obtained in measured and simulation after applying all selection criteria are used as an input for the measurement of the cross sections mitigated for detector effects presented in section 5.5. In the following, the phase space selections applied in this analysis are illustrated.

Muon selections – Due to the highest trigger threshold of 27 GeV in the applied unprescaled single muon trigger (see section 5.1.2.1) the selected muons are required to have at least a corrected reconstructed transverse momentum of 29 GeV for all analysed events. Since the muon system deployed during data taking of the analysed data covered only a region up to $\eta = 2.4$ (see section 4.1.4), the selected muons are required to have $\eta_{\mu} < 2.4$ to ensure the best achievable muon reconstruction with the CMS detector.

Selections on the Dimuon System – Since a dimuon system with an invariant mass close to the mass of a Z boson $m_Z = 91.1876(21)\,\text{GeV}$ [1] is required (see section 5.1.1), at least two muons with opposite electrical charge have to be present in selected events. Next, all combinations of oppositely charged muon pairs are constructed. For all possible combinations of oppositely charged muon pairs the invariant mass of each candidate pair is computed. The mass of a suitable pair is required to be within $71.1876\,\text{GeV} < m_{\mu^+\mu^-} < 111.1876\,\text{GeV}$. When there is more than one pair matching this criteria, the one closest to $m_Z = 91.1876(21)\,\text{GeV}$ is chosen for the subsequent analysis. In addition, the chosen dimuon system's transverse momentum is required to exceed 25 GeV.

A summary of all selection criteria applied on muons and subsequent dimuon systems is shown in table 5.4.

Jet Selections – All fully corrected jets reconstructed from the collection of PF candidates, cleaned for charged PU contributions, clustered using the anti-k_T algorithm with a radius parameter $R = 0.4$, and passing all quality criteria (see sections 4.2.5

Table 5.5: Overview of all final selection criteria (after all quality criteria and corrections) applied to jets in the analysed events. Each of the criteria is applied on the respective identified and corrected collection of reconstructed jets.

Quantity	Selection Criteria
p_T^{jet}	$> 20\,\text{GeV}$
$\lvert y_{\text{jet}} \rvert$	< 2.4

and 5.1.2.2) are required to have a transverse momentum exceeding 20 GeV to further reduce the selection of jets originating from PU. Additionally, the absolute rapidity of the jets is required to be smaller than 2.4. For the subsequent analysis steps only the jet with the highest transverse momentum in the remaining jet collection is relevant.

The jet selections are summarized in table 5.5.

5.1.3 Generator Level and Reconstruction Level Observables

For events generated and simulated in event generators (see section 2.2), two levels of interpretation are defined. For each generated event, the contribution to the measured differential cross sections can be estimated defining a level of "ground truth". It corresponds to the state of the events after generation of the hard scattering, parton shower, hadronization and underlying event, called *generation level* in the following. By subsequent simulation of PU, interactions with the detector, digitization of the detector responses and application of the reconstruction algorithms on the simulated detector signals in the same events their corresponding contribution at *reconstruction level* is established.

The analysed observables at generation level are defined by applying the same definitions and selection criteria as at reconstruction level (see sections 5.1.1 and 5.1.2.4) dropping corrections and quality criteria. Specifically, the correction of triggers, object identification and isolation, and energy and efficiency corrections are not applicable. However, electromagnetic radiation of photons from the muons is corrected at generation level to ensure the same definition of muons at generation and reconstruction level.

5.2 Analysed Data

The data analysed in this work are LHC proton-proton collisions at a center of mass energy of 13 TeV recorded by the CMS experiment in the years 2016 to 2018. This corresponds to an integrated luminosity of approximately $138\,\text{fb}^{-1}$. Only events with at least one muon reconstructed by the trigger are considered.

Due to differences in the detector (see section 4.1), alignment, and other data-taking conditions, the analysed dataset is split into four data-taking periods with distinct re-

construction and calibration. These are labelled 2016preVFP, 2016postVFP, 2017, and 2018. The integrated luminosities corresponding to these four periods are approximately $19.52\,\mathrm{fb}^{-1}$, $16.81\,\mathrm{fb}^{-1}$ [92], $41.48\,\mathrm{fb}^{-1}$ [93], and $59.83\,\mathrm{fb}^{-1}$ [94]. Each period is analysed separately and combined in a later stage of the analysis before the final unfolding.

Due to an issue in the analog pipeline voltage (APV25) readout-chips [95] of the silicon strip tracker of the CMS detector the dataset recorded in 2016 is split into two distinct periods. A slow discharge in the chips at high occupancy in the tracker by high ionizing particles due to more severe pileup conditions than expected led to hits in the tracker not being identified as such. As a consequence, the hit efficiency dropped by up to 10% [96] which lead to unrecoverable inefficiencies in the trigger. The issue was discovered during the operation in 2016. An increase in the preamplifier feedback voltage bias (VFP) was able to recover the hit efficiency of the tracker to normal levels. However, the data recorded previous to this VFP fix was already affected. Therefore, a mitigation of the APV25 issues in the offline reconstruction is applied for the 2016preVFP data.

5.3 Theoretical Predictions

Theoretical predictions of the analysed observables are crucial for the estimation of expected background and signal yields. Furthermore, the simulation of the detector response on the emerging particles in the analysed events is essential for assessing the effects of the detector on the analysed observables. Last but not least, the observables measured in data need to be compared to precise theoretical predictions for tests of the validity of their underlying theoretical models. Therefore, a set of theoretical predictions differing in the utilised technique for their derivation (see section 2.2.2) and the modelled final-state particles are utilised in this analysis.

5.3.1 Event Generators

The first considered class of techniques providing theoretical predictions are derived using MC event generation. With these techniques individual events containing the theoretical counterparts of collision products created at particle collisions, i.e. individual particles, are generated (see section 2.2). This allows to further simulate the detector response of the particles produced in a collision and subsequent steps, i.e. the digitization of the signals and trigger decisions, individually for each event. This enables the estimation of expected signal and background yields at reconstruction level by analysing the simulated events the same way as the data. Furthermore, by comparing the generated events with detector simulation to the same events without, the combined effects of the detector and the reconstruction can be estimated.

For each data-taking period, an individual statistically independent set of simulated

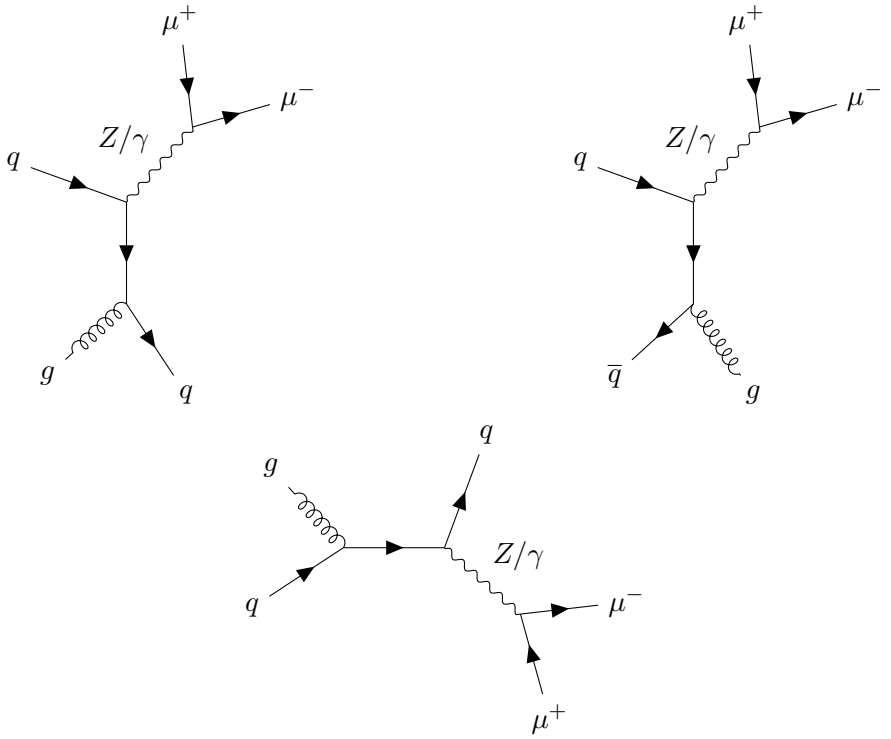
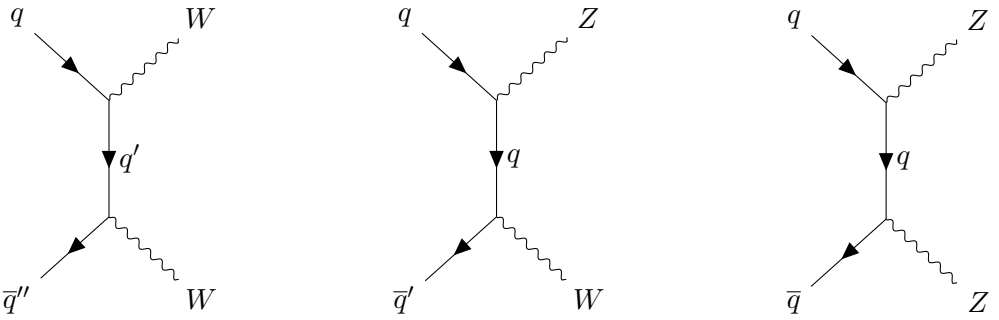


Figure 5.3: Example tree-level Feynman diagrams of the signal process creating a signature with a pair of oppositely charged muons and at least one parton creating a jet in the final state.

events is generated by the CMS collaboration involving the simulation of the respective state of the CMS detector.

Signal – Signal events are modelled by generating proton-proton collisions at 13 TeV center-of-mass energy producing a pair of oppositely charged muons inclusive in the number of additional jets. The LO Feynman graphs producing such a signature are depicted in fig. 5.3. For the signal sample MadGraph5_aMC@NLO [97, 98], implementing the MC@NLO matrix element to parton shower matching method [30], is used to generate events containing a pair of oppositely charged leptons with an invariant mass > 50 GeV and up to two partons at NLO matrix element accuracy in QCD respectively. The production of unstable tau-lepton-antilepton pairs and all their decays are included in this sample. The leptonic decay of the tau-lepton-antilepton pair into two muons and neutrinos creates a signature which contributes to the selected signal region. All events are further processed by Pythia 8 [29], that generates the parton shower and models the hadronization and underlying event for full particle level event generation resulting in a sample inclusive in the number of jets below the fragmentation scale. The individual contributions of each exclusive parton multiplicity are merged using the FxFx method [99]. The event weights of the generated signal events are normalized to match the cross



(a) WW **(b)** WZ **(c)** ZZ

Figure 5.4: Tree-level Feynman diagrams of di-boson background processes with similar signature as the $Z(\rightarrow \mu\mu) + \text{jet}$ signal process. Example Feynman diagrams for the production of two W bosons fig. 5.4a, a W boson and a Z boson fig. 5.4b, and two Z bosons fig. 5.4c are shown.

section prediction of $6077.22\,\text{pb}^{-1}$ for the Drell-Yan process at NNLO accuracy in QCD and NLO accuracy in EW obtained with FEWZ [100–103] for the fiducial phase space. This calculation utilises the same PDF set NNPDF 3.1 [43] as in the generated signal sample. This corresponds to an expected yield of approximately 98.48% of the total selected events.

Backgrounds – Processes, which produce event signatures That mimic the selected signal signature, need to be taken into account in the analysis by correcting for their contribution to the analysed observables. These signatures arise from misidentification of particles due to limitations in the detection and reconstruction or due to particles leaving the acceptance region of the analysis and the detector. Processes, which create the same final states as the signal but are not targeted in the analysis, are called *irreducible backgrounds*. These cannot be eliminated through experimental techniques and need to be subtracted in the analysis. The background processes, are the production of fermions via EW processes involving multiple bosons and the production of top-quarks in association with jets and fermions.

The production of multiple leptons in association with jets is modelled via the production of EW boson pairs decaying into fermions using Pythia 8 [29] at LO accuracy. The production cross section of two EW-bosons at the LHC is, however, small compared to the production of single bosons and QCD partons. Consequently, although the signatures of these background processes resemble the signal's, they contribute the least with a total expected event yield of approximately 0.39% in the selected phase space. There are three classes of di-boson production contributing to this analysis. Example Feynman graphs for the production of boson pairs at tree-level are depicted in fig. 5.4.

For a pair of Z bosons both decaying into fermions multiple combinations of misidentifica-

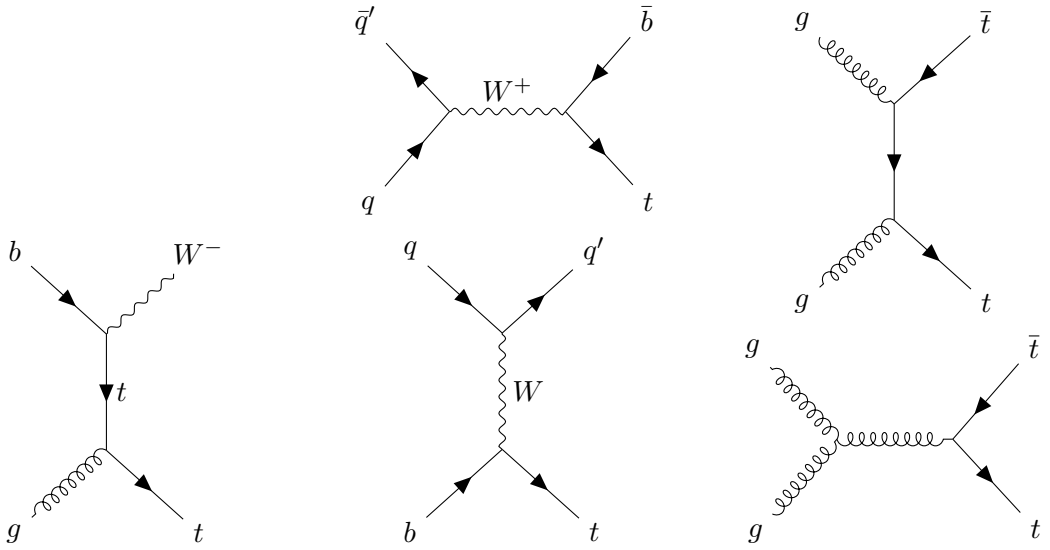
tion exist. When two of the daughter fermions correspond to a pair of oppositely-charged muons and the other fermions are quarks producing jets in the selected kinematic region, the events' final states contain the same particles as in the signal process and compose therefore an irreducible background. When the second fermion pair are neutrinos or charged leptons, additional hadronic activity in the event or tau decays can still lead to a reconstruction of at least one jet selecting this kind of background event. In the case, that none of the Z bosons decays into a pair of muons the background is strongly suppressed by the high accuracy of the reconstruction of muons in CMS (see section 4.2). Nevertheless, this remaining background is modelled and included in the analysis. A cross section of $12.17 \pm 0.02\,\text{pb}$ for this sample is obtained from Pythia 8 at LO accuracy. The uncertainty includes the statistical uncertainty related to the limited amount of samplings in the MC integration only. The channel contributes with approximately 0.17% to the expected event yield.

W bosons decay either into two quarks or a charged lepton and the corresponding neutrino. When two W bosons decay into muons, additional hadronic activity that results in a jet is needed for an event to be selected in this analysis. When two of the leptonically decaying W bosons produce tau-leptons, which further decay into muons and neutrinos, a similar signature as the signal is created if additional hadronic activity is reconstructed as a jet. Events, where one W boson decays into quarks and the other into leptons require a misidentification of one of the products as a muon. This process is suppressed by the accurate reconstruction of muons in CMS. The cross section for the production of a pair of W bosons of $\sigma_{\text{WW}} = 118.7^{+2.5\,\%}_{-2.2\,\%}\,\text{pb}$ is calculated in [104]. The uncertainty is estimated by varying the factorization and renormalization scales by factor of two up and down and variations of the PDF parameters. This channel contributes only with approximately 0.03% to the total number of expected events in the selected phase space.

A pair of a W boson and a Z boson create a signal-like signature if the W boson decays into quarks and the Z boson into a muon pair. All other decay channels are again suppressed by CMS's purity with respect to the reconstruction of muons. Pythia 8 calculates a cross section of $27.6 \pm 0.4\,\text{pb}$ for this channel. The uncertainty corresponds to the statistical uncertainty associated with the limited amount of sampled events. This channel contributes 0.19% to the expected event yield.

The production of top-quarks and their decays into leptons and jets is responsible for the largest background in the analysis due to the high production cross section of top-quarks compared to di-bosons and a signature of the decay products similar to the signal process. This leads to a contribution of top-quark production modes of approximately 1.13% to the expected event yields in the selected phase space. Example Feynman graphs for the production of top-quarks at tree-level are shown in fig. 5.5.

The production of top-quark pairs and their decay into quarks and leptons using a narrow-width approximation is simulated with POWHEG [31, 105, 106] at NLO accuracy in perturbative QCD with the implementation described in [107] and interfaced to



(a) tW **(b)** tq **(c)** $t\bar{t}$

Figure 5.5: Example Feynman diagrams at tree-level for the production of top-quarks with similar signatures as the $Z(\rightarrow \mu\mu)$ + jet signal process. Example Feynman diagrams for the production of single top-quarks in association with a W boson fig. 5.5a, a single top-quark in association with additional partons fig. 5.5b and top-quark-antiquark pairs fig. 5.5c are shown.

Pythia 8 [29]. Top-quarks decay almost exclusively into a b-quark and either quarks or a charged lepton and neutrino via an EW-decay. With both top-quarks decaying into b-quarks (reconstructed as jets), muons, and neutrinos a signature similar to the signal's is created since the neutrinos pass the detector undetected. All other decay channels are suppressed since they would require a misidentification of at least one decay product as a muon. Therefore, only the contribution from the decay channel producing two muons is considered in this analysis. The yields of the generated events are normalized to match the prediction for the cross section of top-quark-antiquark pair production at the LHC of 831.76 pb obtained from a calculation at NNLO accuracy in perturbative QCD including the resummation of up to next-to-next-to-leading-logarithmic (NNLL) infrared gluon contributions assuming a top-quark mass of 172.5 GeV with Top++2.0 [108]. The cross-section value is multiplied by the branching ratio for the dileptonic decay into muons resulting in a cross section of 88.29 pb. A relative uncertainty of $+4.8\%$ and -6.1% is derived from PDF variations and variations of the strong coupling constant α_s using the PDF4LHC prescription [109, 110] with the MSTW2008NNLO [111], NNPDF2.3_-5F [112], and CT10NNLO PDF sets [42]. A nominal contribution of this production mode to the event yield of approximately 0.97% is expected.

Single top-quarks are expected to be predominantly produced at the LHC in association with a W boson or quarks. Their production and the decay of the top-quark (and W boson) is modelled with POWHEG at NLO accuracy in perturbative QCD. The as-

sociated production with a W boson is implemented following [113]. For the top-quark and W boson both decaying into a muon plus leptons and quarks a similar signature as the signal's is created. The production of a top-quark in association with other quarks is implemented as described in references [114, 115]. Here, the decay of both the top-quark and an EW decay of an associated heavy-flavour quark can produce muons that lead to a signal-like signature. Other decay channels are suppressed because of the CMS detector's sensitivity to muons. The cross sections for the associated production with a W boson of $71.1\,\text{pb} \pm 3.8\,\text{pb}$ is obtained from calculations at NNLO accuracy [116, 117]. The cross section for the single top-quark (top-antiquark) production in association with quarks of $\sigma(tq) = 145.0^{+2.8}_{-1.9}\,\text{pb}$ ($\sigma(\bar{t}q) = 87.2^{+1.8}_{-1.5}\,\text{pb}$) is computed at NNLO QCD using a top-quark mass of 172.5 GeV in [118]. The uncertainties include variations of the factorization and renormalization scales by a factor of two and combined PDF and α_s variations according to [119]. Due to the lower production cross section for the production of single top-quarks compared to the production of top-quark-antiquark pairs, the small branching ratio of the relevant decay channels, and the phase space coverage of the decay products the production of single top-quarks is expected to contribute approximately 0.16% to the event yield in the selected phase space.

5.3.2 Fixed-Order Calculations for the Signal Process

For the interpretation of the measured cross sections theory predictions with the highest achievable precision are preferred. The most precise cross section predictions in perturbation theory are obtained by including higher order terms into the perturbative series (see section 2.2.2). For most scattering processes, the necessary matching procedure in event generation (see section 2.2.2.3) limits the inclusion of higher order terms into the ME calculations. Therefore, the predictions at the highest orders in perturbation theory are obtained in calculations which do not include the effects generated by PS and non-perturbative models but provide predictions with the lowest obtainable uncertainties related to the unknown contribution of missing higher-order terms in the perturbative series.

Since these predictions are compared to unfolded results, which mitigate detector effects, no distinction between individual data taking periods is made. Consequently, a single set of predictions is sufficient.

Recent developments allow calculating the fully differential cross sections of the production of pairs of oppositely charged muons plus partons at the LHC with NNLO accuracy in perturbative QCD using [120, 121]. However, these elaborate calculations are not available in time with sufficiently small statistical uncertainties for this thesis. Comparisons to the predictions are therefore left for future studies.

5.3.2.1 Elektroweak Corrections

A first estimation of the convergence of the perturbative series can be obtained by studying the individual terms in this series in orders of the strong and EW couplings. The contribution to the cross section of the signal process added by terms in NNLO QCD is expected to be of the same magnitude as the contribution added by NLO EW terms in selected regions of phase space. Consequently, the corrections added by NLO EW terms need to be added to the calculation. However, they are not available in time for this thesis. It is left for future studies to include the effects of NLO EW terms to the fixed-order calculations used in the comparison with the measured cross sections.

5.3.2.2 Non-Perturbative Corrections

The fixed-order calculations are computed approximately assuming that the transmitted energy in the hard collision is much bigger than the scale of QCD Λ_{QCD} at which bound-states are formed (see section 2.2.2). However, this assumption does not hold over the whole selected phase space. In certain kinematic regions, contributions to the observed cross sections by ME terms evaluated at scales approaching Λ_{QCD} are expected. There the perturbative calculation becomes unstable and corrections need to be applied.

Derivation – The corrections can be derived using event generators which implement non-perturbative (NP) models to account for such effects (see section 2.2.3). By comparison of predictions on the relevant observables made by the event generators with and without the NP effects included in the generation chain (see section 2.2) correction factors c_{NP} can be estimated. From the division of the two predictions X denoting the obtained distribution with full generation chain including the non-perturbative effects and Y denoting the distribution of the observable obtained from the partial event generation the c_{NP} can be derived as

$$c_{\mathrm{NP}} = \frac{X}{Y} \,. \tag{5.11}$$

As a consequence, the correction factor c_{NP} needs to be estimated from a random distribution following the ratio of the random distributions. A high amount of generated events in both generation scenarios are required, since both nominator and denominator follow a random distribution that can be approximated by a normal distribution with expectation value μ equal to the predicted value and variance σ^2 for large numbers of generated events. Also, the variance of this normal distribution is inversely proportional to the number of events (see section 2.2.1) minimizing the uncertainty for large numbers of generated events. A small uncertainty is strictly necessary for the estimation of the correction factors. Since the distribution describing the ratio of two normal distributed random variables has in general no statistical moments due to divergent tails [122, 123] it cannot be determined without an approximation. For small coefficients of variations for the denominator distribution $\delta_Y = \frac{\sigma_Y}{\mu_Y} < 0.1$ the distribution of the quotient can be

approximated as another normal distribution with mean

$$c_{\mathrm{NP}} = \frac{\mu_X}{\mu_Y} \tag{5.12}$$

and variance

$$\sigma_c^2 = \frac{\sigma_X^2}{\mu_Y^2} + \frac{\sigma_Y^2 \mu_X^2}{\mu_Y^4} \tag{5.13}$$

as shown in [124]. This can be directly computed from the statistical moments of the nominator and denominator distributions.

For this purpose, more than 10^9 events have been created for each considered scenario (one full and three partial generation chains) using MadGraph5_aMC@NLO [97, 98] for the generation of the hard scattering interfaced to Herwig [27, 28]. The generation of the parton shower (PS), hadronization and underlying event (UE) are generated by Herwig. The hard scattering was configured to produce proton-proton collisions at a center-of-mass energy of 13 TeV producing a pair of oppositely charged muons and a jet at LO and a separate set of events at NLO perturbative QCD, respectively, matched to the Herwig PS using the MC@NLO matching method. Unlike Pythia 8, Herwig implements an angular ordered parton shower (see section 2.2.2.2) as default and a cluster hadronization model (see section 2.2.3.1). The corresponding configuration files used for the production of the full and partial scenarios at LO and NLO accuracy in perturbative QCD are shown in appendix A.1.1.

In these generation chains, the hadronization and UE models are non-perturbative. The hadronization procedure includes no perturbative components and is based purely on empirical models motivated by perturbative QCD. The UE models in Herwig add additional QCD interactions to the event originating from MPI. However, their effective kinematic structure and mixing to the original hard interaction are based on empirical models. Therefore, the NP-corrections are defined by plugging the observables obtained from the full generation chain, including the hard interaction, PS, hadronization and MPI models, denoted as X = ME+PS+Had+MPI, and the observables obtained from the partial generation chain including only the hard interaction and PS, denoted as Y = ME+PS, into eq. (5.11).

To obtain the relevant observables for this analysis, namely the cross sections differentially in bins of y_b, y^*, and p_T^{Z} (see section 5.1.1), the analysis steps are repeated on the events on generator level (see section 5.1.3) for all scenarios. For this purpose a Rivet [125] routine (see appendix A.1.2) is utilised which runs directly on the output of the generator. From the resulting cross sections at generator level the non-perturbative correction factors c_{NP} are derived following eq. (5.11).

Smoothing of Statistical Fluctuations – The obtained correction factors c_{NP} are subject to statistical fluctuations due to the limited number of generated events for both

the full and partial generation chains. Therefore, to estimate the correction factors in a limit of infinite numbers of generated events, a parametric function of p_T^Z is fitted to the corresponding correction factors for each y_b-y^*-bin. The chosen function modelling the parametric behaviour of the correction factors in $x = p_T^Z$

$$f(x; a, b, c) = ax^b + c \tag{5.14}$$

contains three freely floating parameters a, b, and c. The parameters are fitted to the obtained correction factors using the least squares objective function

$$p(a, b, c) = \frac{\hat{y} - f(\hat{x}; a, b, c)}{\sigma_y} \tag{5.15}$$

with $\hat{y} = c_{\mathrm{NP}}$, and $\sigma_y = \sigma_c$ as defined in eqs. (5.12) and (5.13), and $\hat{x}$ defined as the center of the corresponding p_T^Z bin, by minimizing eq. (5.15) with respect to its free parameters a, b, and c. The minimization is performed numerically using either the Nelder-Mead-method [126, 127] or a trust region algorithm [128], whichever converges better. When a parameter cannot be constrained due to too many degrees of freedom in the model function the parameter c in eq. (5.14) is set to 1 and the minimization is repeated. When this fails parameter b in eq. (5.14) is set to 1 and the fit is performed another time. The uncertainty of the fit is derived by evaluating

$$\sigma_f^2 = J^T C J \tag{5.16}$$

with the inverse Hesse matrix evaluated at the best fit value C multiplied by the Jacobian vector of eq. (5.14) with respect to its model parameters J and its transposed J^T.

Non-perturbative Effects – The original correction factors and the fitted smoothing function with uncertainty σ_f for each y_b-y^*-bin derived at LO and NLO accuracy in QCD are shown in appendix A.1.3 respectively. With the presented fitting procedure all fits converge without issues. The smoothed correction factors are shown in fig. 5.6 for the generations at LO and NLO accuracy in perturbative QCD for comparison.

The effect of non-perturbative models on the expected differential cross sections is the largest for small p_T^Z and diminishes for high p_T^Z. For small y^* the correction factors are smaller than zero and do not exceed minus 10%. However, the non-perturbative effects on the cross sections grows in absolute values with increasing y^* exceeding zero at $1.0 < y^* < 1.5$ and reaching its maximum in the last y^*-bin. In contrast, no significant dependence of the non-perturbative effects on y_b can be observed.

An additional dependency of the non-perturbative models on the measured cross sections can, however, be observed on the order or perturbative accuracy in QCD. For the generation at NLO accuracy the correction factors are significantly smaller in absolute size than at LO. To investigate the reason for this dependency the non-perturbative steps in the generation are studied separately.

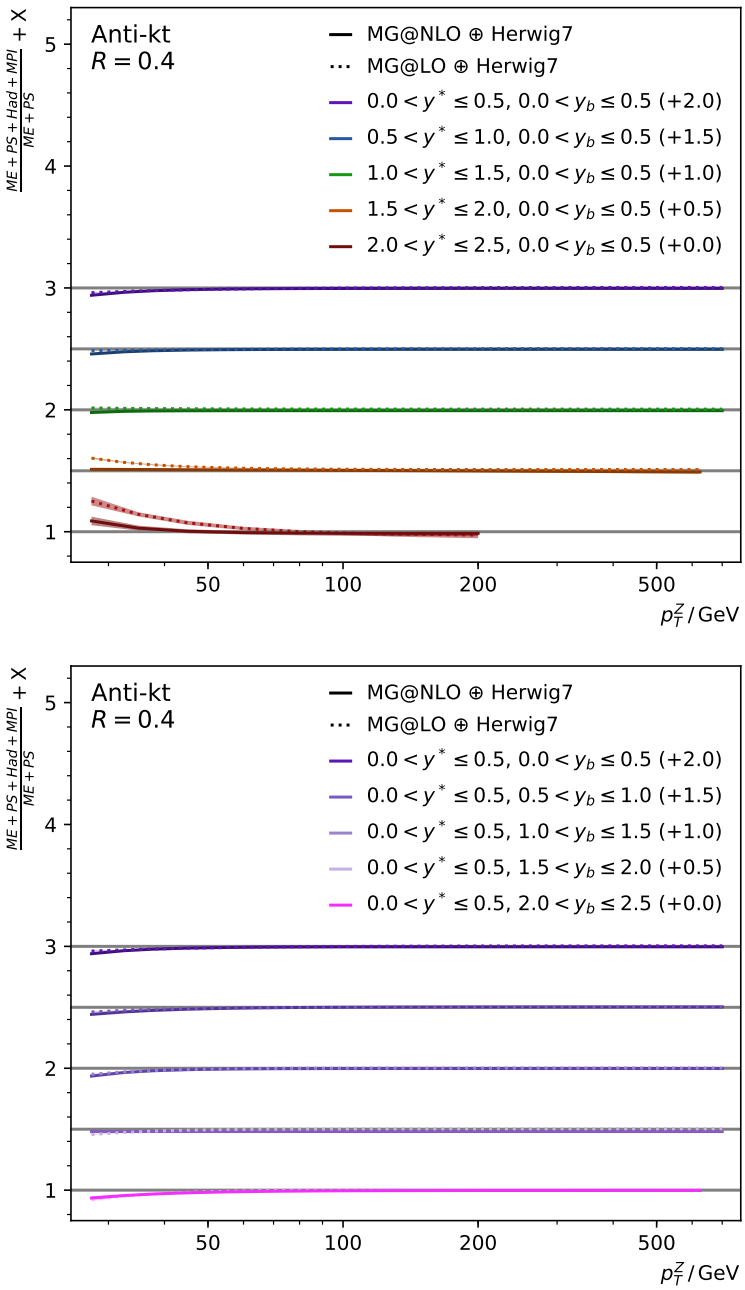


Figure 5.6: Comparison of smoothed non-perturbative correction factors at LO and NLO accuracy for jets clustered with anti-k_T with radius parameter $R = 0.4$. Only the y_b-y^*-bin with constant $0 < y_b < 0.5$ and increasing y^* (top) and constant $0 < y^* < 0.5$ and increasing y_b (bottom) are shown.

Hadronization and MPI Effects – The effects of hadronization and MPI are individually studied by replacing the observables obtained in the full generation chain by the ones obtained from the partial generation including the hard scattering, PS and hadronization, denoted as X = ME+PS+Had, or hard scattering, PS and MPI, denoted as X = ME+PS+MPI, in the evaluation of eq. (5.11) respectively. The obtained correction factors, smoothed and original, are shown for all bins and both examined orders in perturbative QCD in appendix A.1.3. The smoothed correction factors for selected y_b-y^*-bins in both LO and NLO for hadronization effects are shown in fig. 5.7, and for MPI in fig. 5.8.

It can be observed in fig. 5.7 that there is neither a dependency of hadronization effects on the measured cross sections in y_b, y^* nor the perturbative order in QCD in the modelling of the hard interaction. The effects of hadronization are stable in all except the transverse momentum of the dimuon system p_T^Z matching the expectation of non-perturbative effects reducing for incrementally harder regions of phase space. The effect of hadronization on the cross section is in general negative and approaches a value of no effect for high p_T^Z. Consequently, neither the observed increase of the non-perturbative corrections with y^* nor the perturbative order is subject to hadronization effects.

Figure 5.8 shows the isolated effect of MPI on the cross section. It can be observed that for small y^* the correction factors induced by MPI do not change with y_b. There is only a tiny effect positive in sign for small p_T^Z visible approaching zero effect for high p_T^Z. The same trend in p_T^Z is visible in all y_b-y^*-bins. However, with increasing y^* the overall scale increases. Also, the trend is more pronounced for the generation at LO accuracy compared to NLO. The increase of the correction factors with y^* and decrease with perturbative order are the same trends as observed for the full non-perturbative corrections indicating that the origin of the observed effects lies in the MPI modelling.

It is unclear, why the MPI has such a dependency on y^*. It can be argued, that in the forward region MPI contributions are generally larger due to the scale of MPIs being smaller than the hard interaction leading to a smaller population in the transverse direction of the beam. With high y_b, however, most activity from the hard interaction is oriented towards one side leaving the other hemisphere to be populated by the UE. With high y^*, no such bias towards a clear direction is expected. Consequently, the UE is expected to populate the phase space more isotropic but the products of the hard interaction populate the phase space preferrably with high rapidities and pick up more contributions by the UE. However, further studies for verification of this theory are required. One suggestion would be to measure the UE in bins of y^* similar to [129].

On a similar level, the observed trend with the perturbative order used for the modelling of the hard interaction can be motivated. For LO only one jet is produced in the hard interaction. Therefore, one hemisphere of the event's phase space is not filled by coloured particles leaving a large phase space for MPIs to fill. As a consequence of the higher MPI activity, more jets originating from MPI are selected and the correction

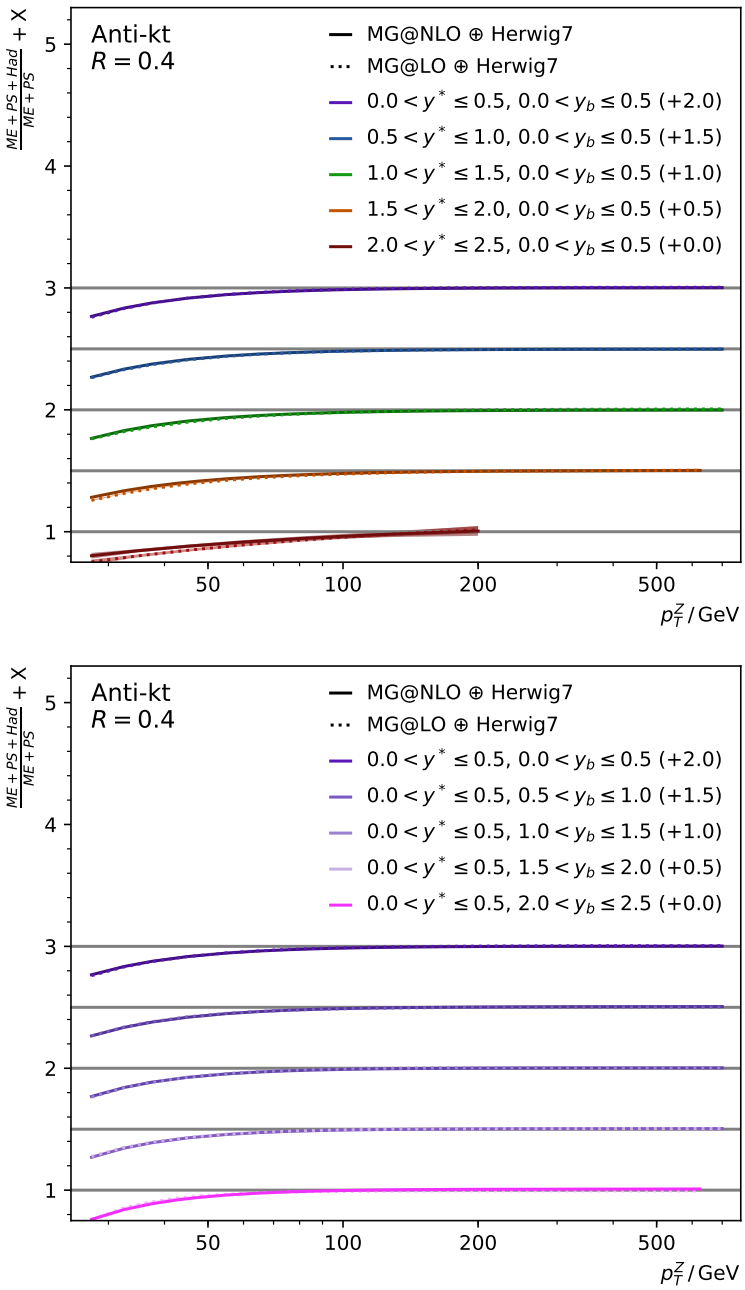


Figure 5.7: Comparison of smoothed hadronization correction factors at LO and NLO accuracy for jets clustered with anti-k_T with radius parameter $R = 0.4$. Only the y_b-y^*-bin with constant $0 < y_b < 0.5$ and increasing y^* (top) and constant $0 < y^* < 0.5$ and increasing y_b (bottom) are shown.

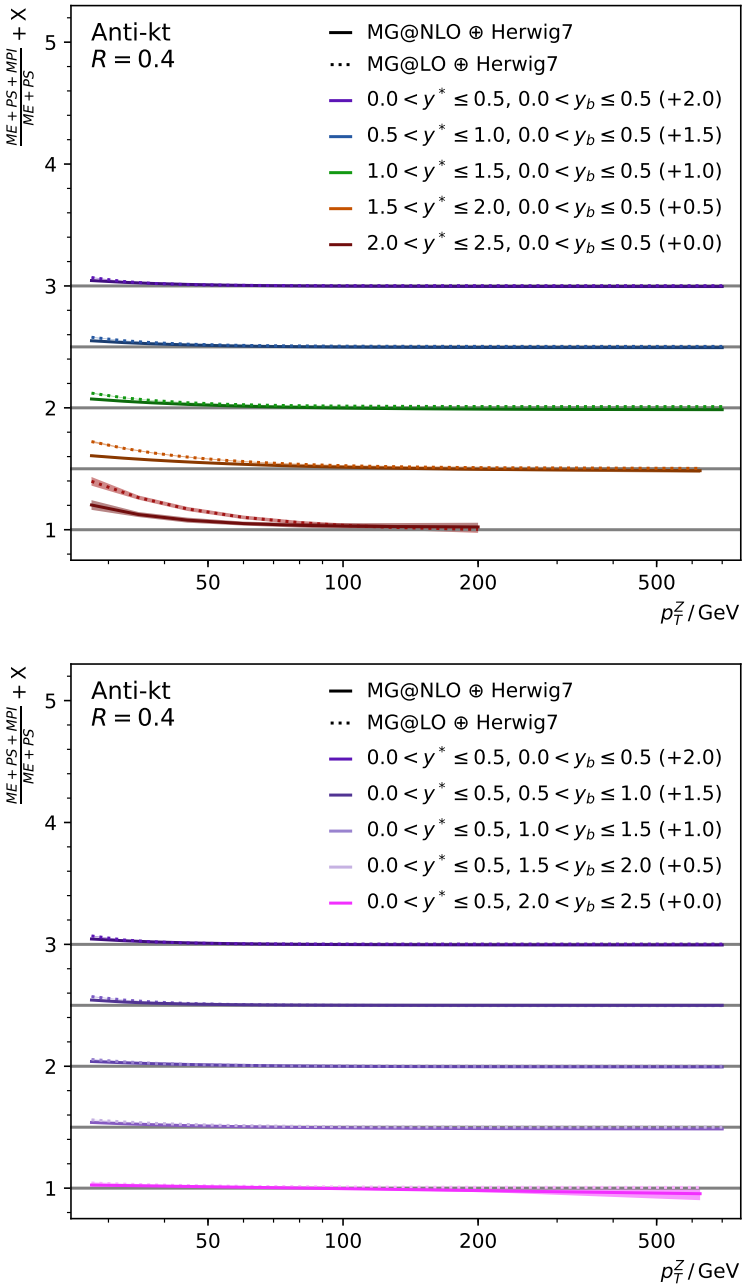


Figure 5.8: Comparison of smoothed MPI correction factors at LO and NLO accuracy for jets clustered with anti-k_T with radius parameter $R = 0.4$. Only the y_b-y^*-bin with constant $0 < y_b < 0.5$ and increasing y^* (top) and constant $0 < y^* < 0.5$ and increasing y_b (bottom) are shown.

factors are larger. At NLO accuracy, a second jet is generated in the hard interaction leaving less phase space available for MPIs. Therefore, the effect introduced by MPIs is smaller. Accordingly, with predictions at higher orders than NLO the effect of MPIs would diminish further. Further studies are needed, however, to confirm or reject this hypothesis.

5.4 Combination of Datasets and Scrutiny

All selections, corrections and quality criteria (see section 5.1) are applied individually on each set of data measured in one of the four data-taking periods (see section 5.2) and the corresponding simulated sets of data for background and signal processes (see section 5.3). This is necessary to account for differences in the detector and accelerator state during the respective period. For this purpose, the event yields in the simulation are weighted with the corresponding cross sections of the simulated processes and luminosities of the corresponding data-taking period on top of the correction and reconstruction weights. Since both the weighted events in the data and simulation are independent of each other the corresponding statistical uncertainties and yields are given by the statistics of weighted Poisson events [91] for each of the individual data-taking periods respectively. The same holds for combination of the data of the individual periods into a single set given that there are no significant systematic differences observed between the data-taking periods.

Compatibility between Data-Taking Periods – For scrutiny of the analysis workflow the event yields in data are compared to the expected yields extracted from the combined yields of selected events in background and signal simulations for each data-taking period for several observables, following the procedure described in [89]. The event yields differential in bins of

- the pseudorapidity of the two selected oppositely charged muons are shown in figs. 5.9 and 5.10,
- the azimuth angle of the two selected oppositely charged muons are shown in figs. 5.11 and 5.12,
- transverse momentum of the two selected oppositely charged muons are shown in figs. 5.13 and 5.14,
- the rapidity of the dimuon system are shown in fig. 5.15,
- the azimuth angle of the dimuon system are shown in fig. 5.16,
- the invariant mass of the dimuon system are shown in fig. 5.17,
- the pseudorapidity of the hardest jet are shown in fig. 5.18,
- the azimuth angle of the hardest jet are shown in fig. 5.19,

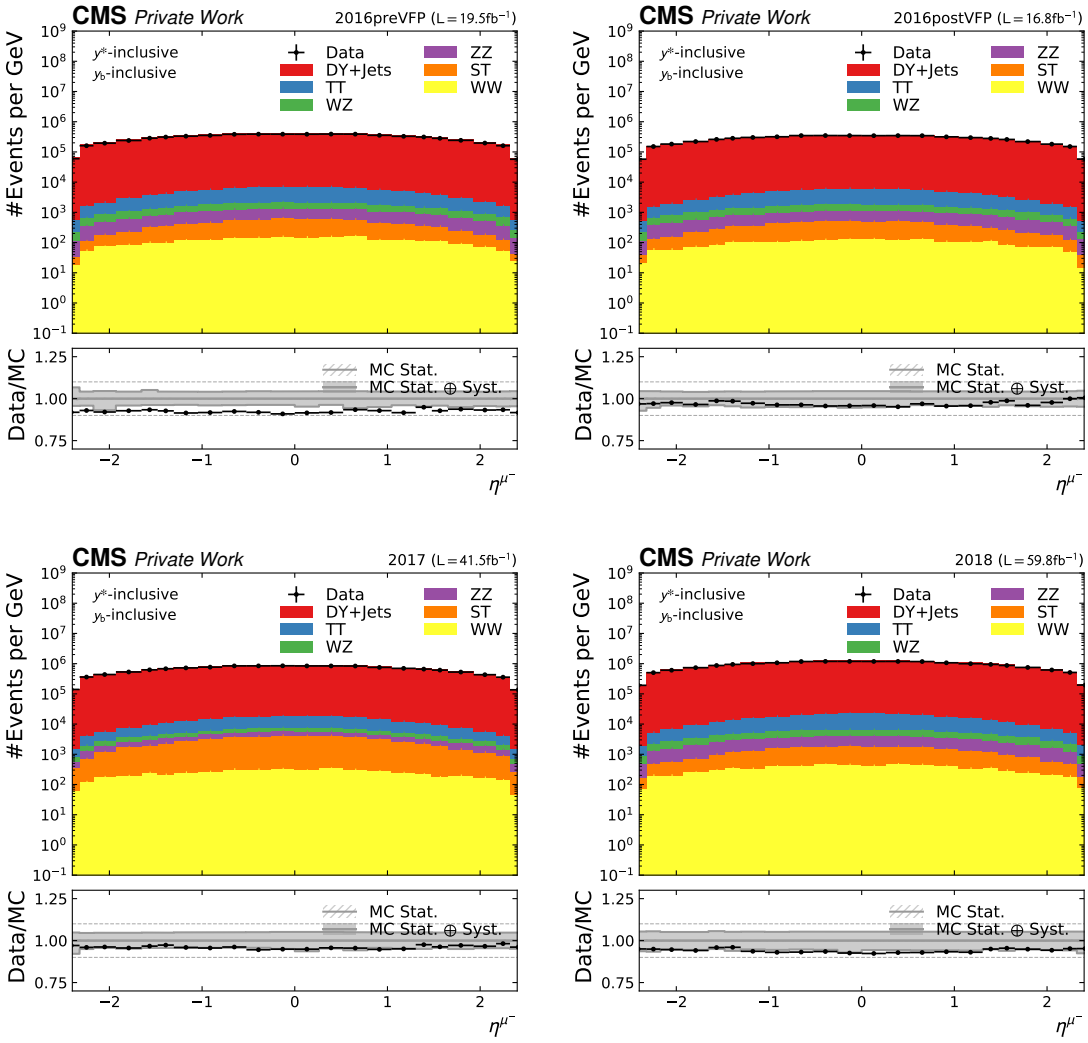


Figure 5.9: Comparison of the pseudorapidity of the negatively charged muon selected for the dimuon system reconstruction η^{μ^-} at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

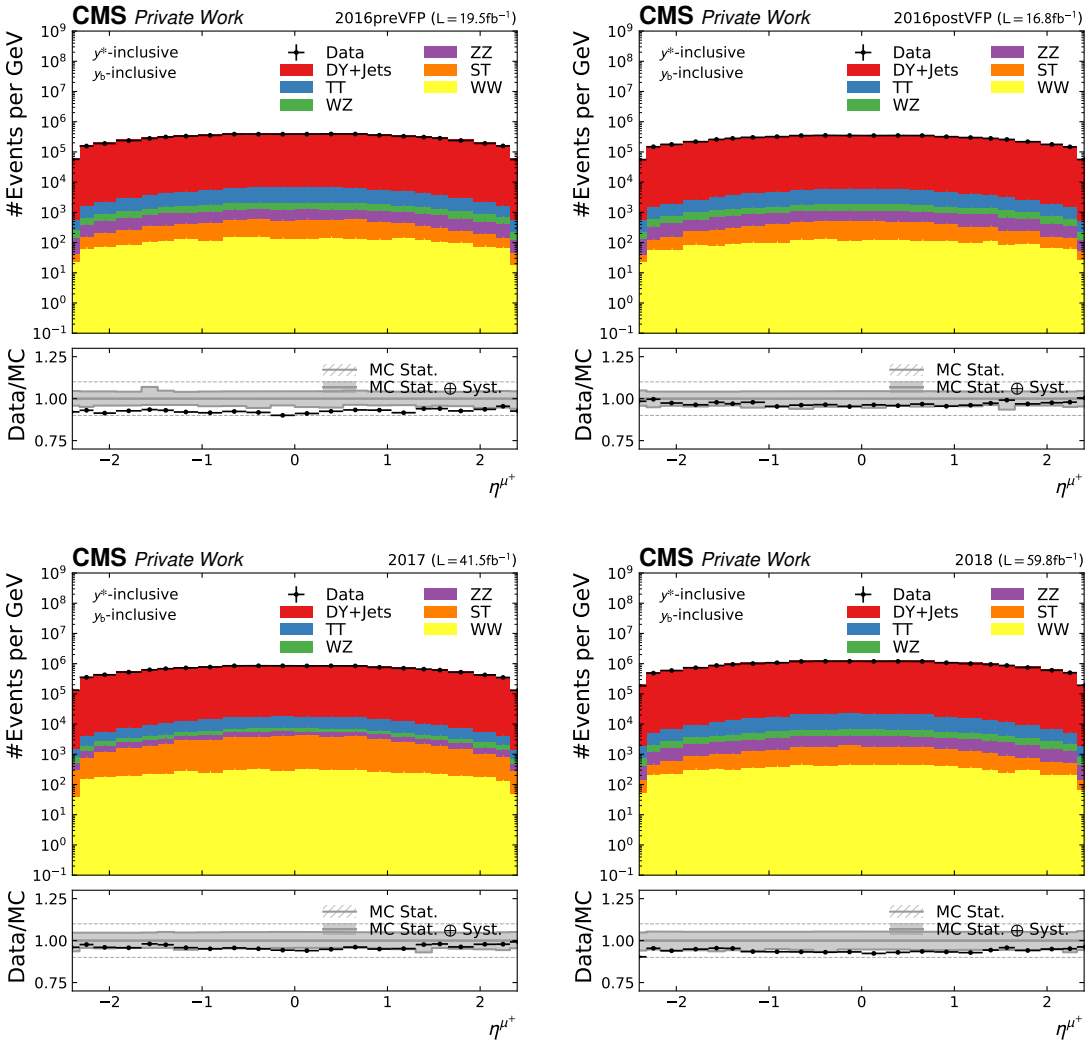


Figure 5.10: Comparison of the pseudorapidity of the positively charged muon selected for the dimuon system reconstruction η^{μ^+} at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

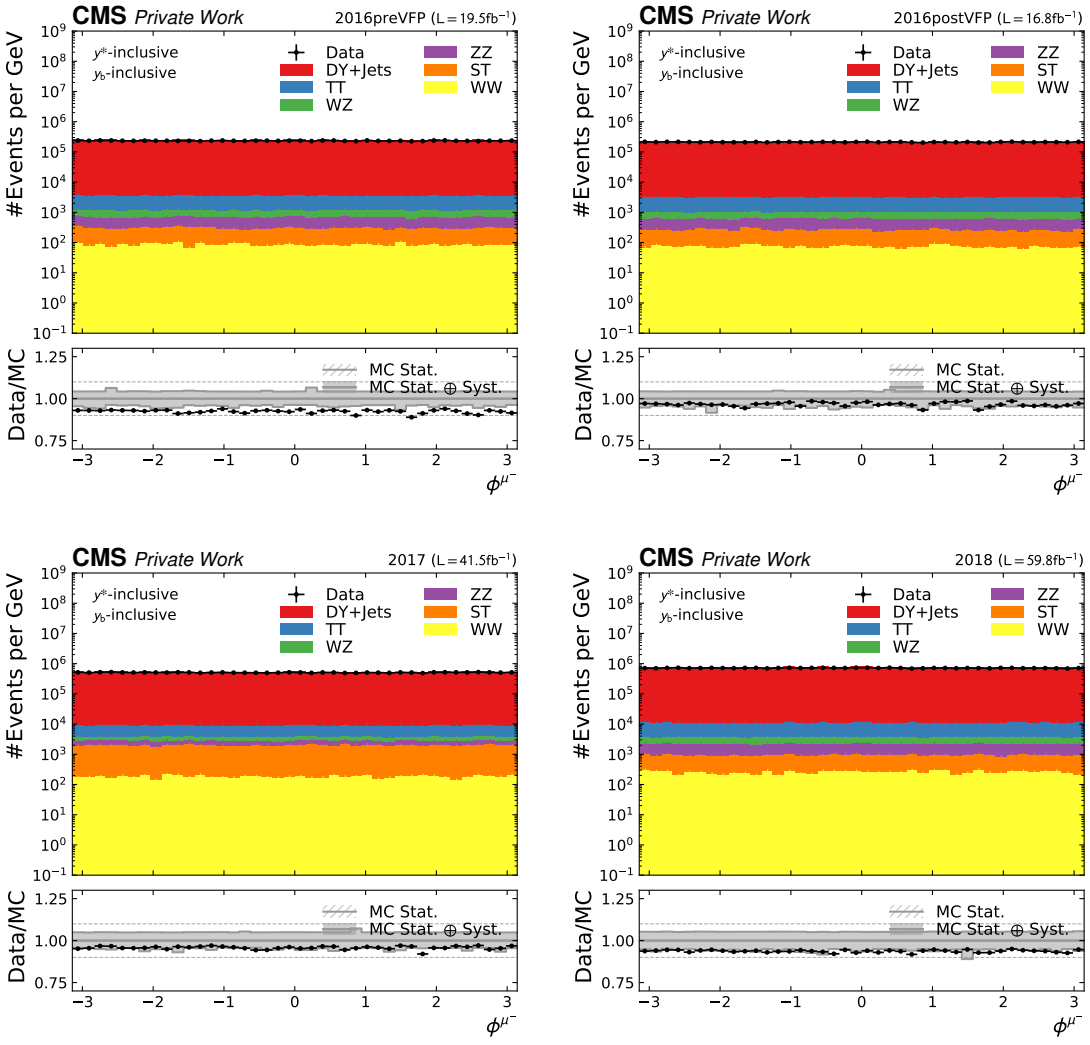


Figure 5.11: Comparison of the azimuth angle of the negatively charged muon selected for the dimuon system reconstruction ϕ^{μ^-} at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

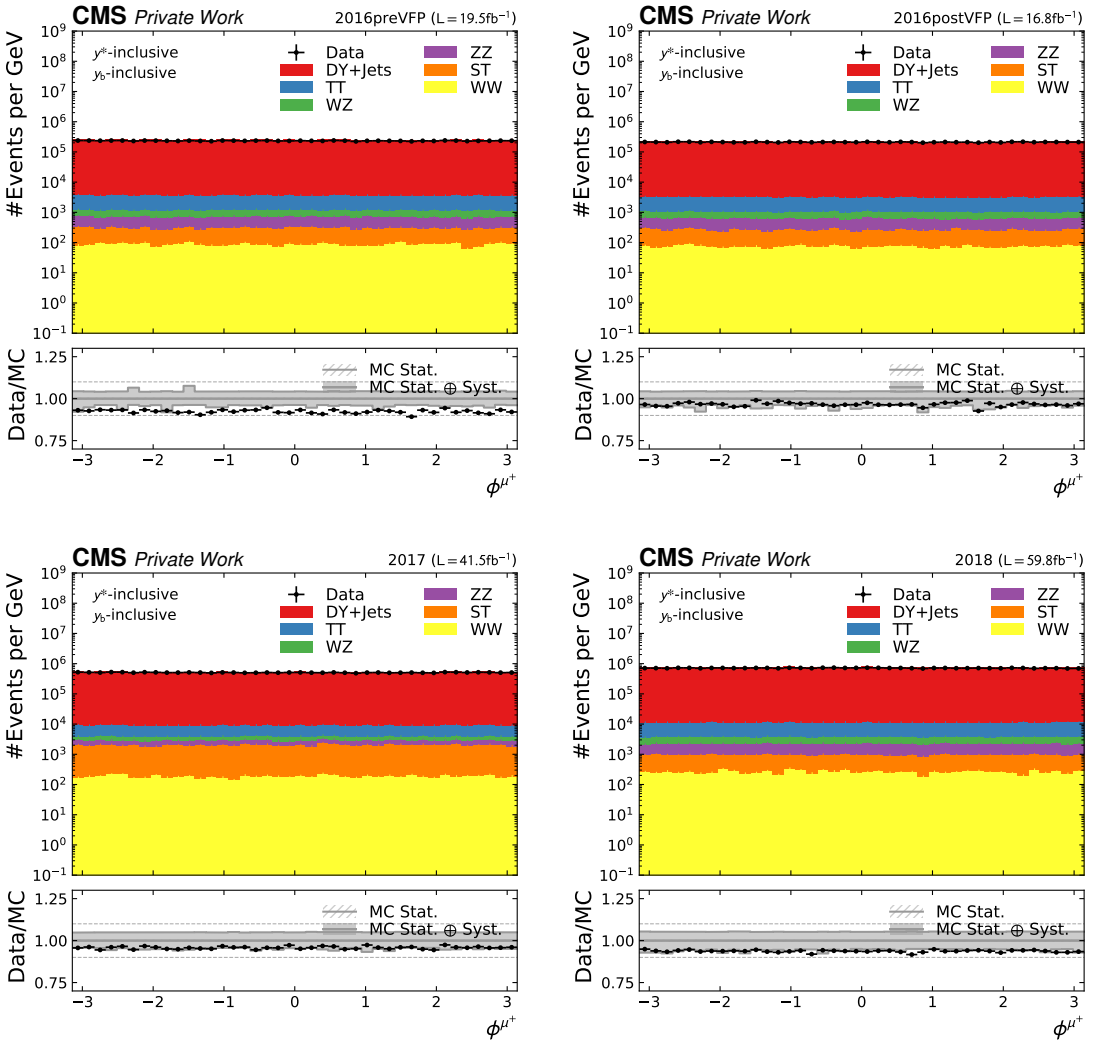


Figure 5.12: Comparison of the azimuth angle of the positively charged muon selected for the dimuon system reconstruction ϕ^{μ^+} at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

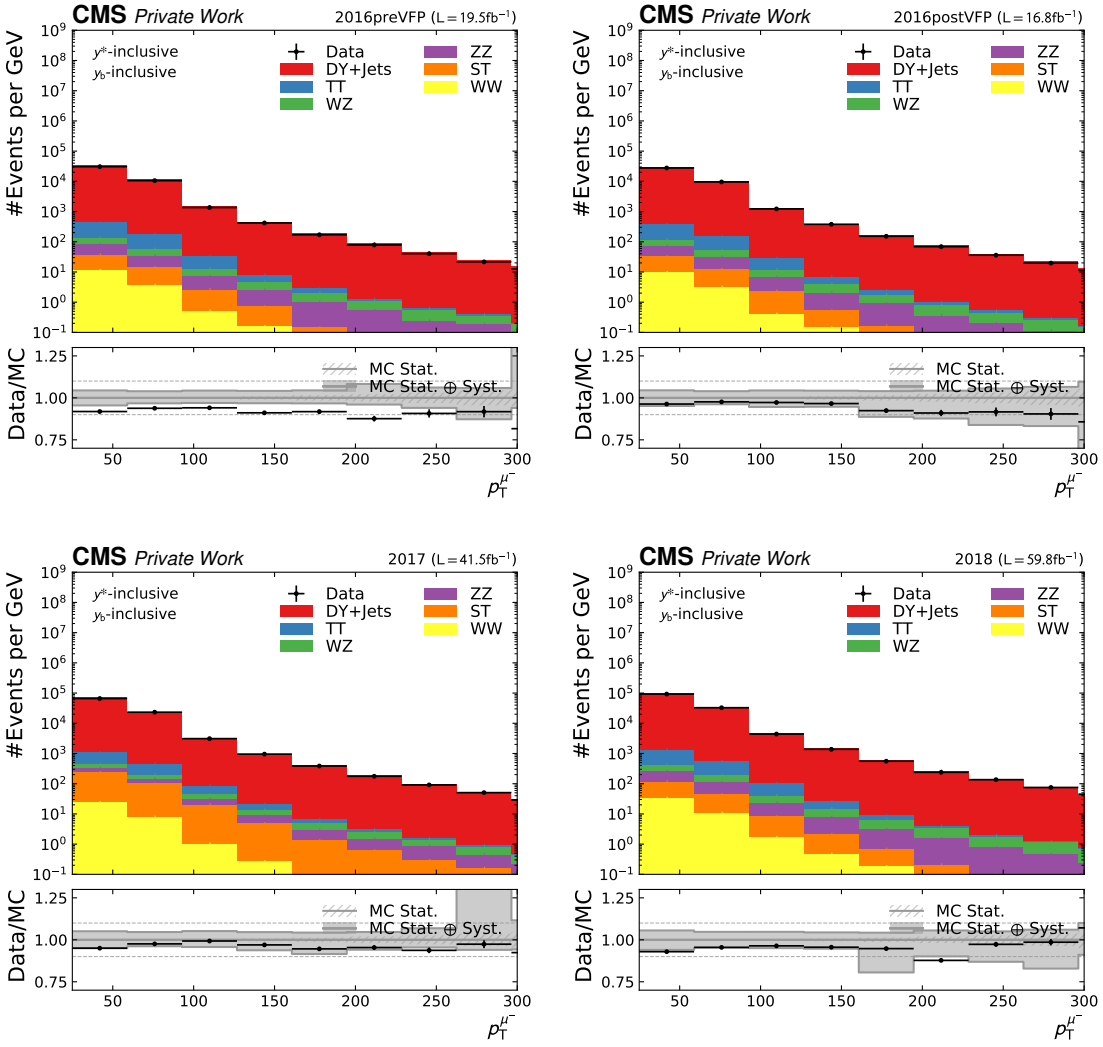


Figure 5.13: Comparison of the transverse momentum of the negatively charged muon selected for the dimuon system reconstruction $p_T^{\mu^-}$ at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

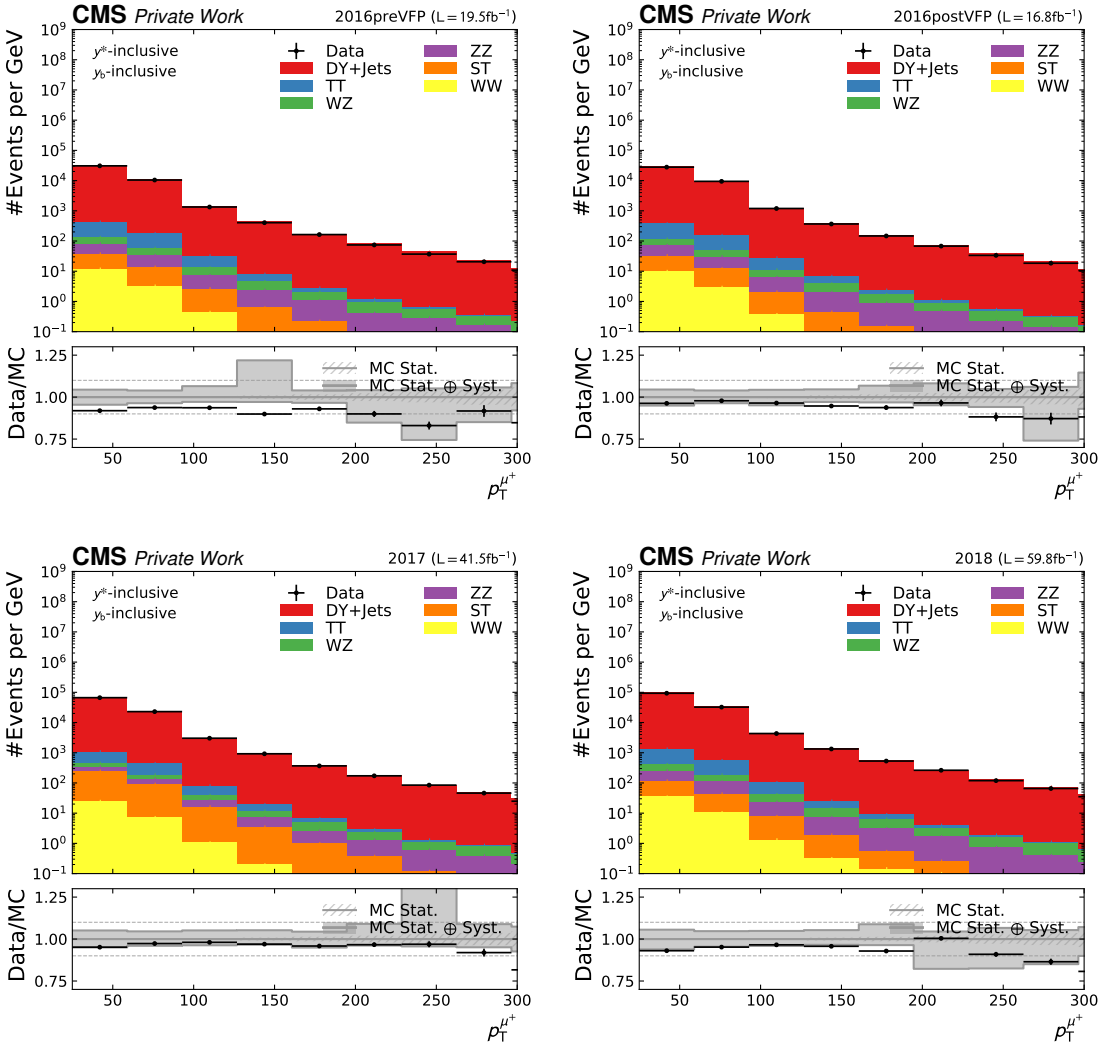


Figure 5.14: Comparison of the transverse momentum of the positively charged muon selected for the dimuon system reconstruction $p_T^{\mu^+}$ at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

- and the transverse momentum of the hardest jet are shown in fig. 5.20,

respectively for all four data-taking periods.

In all those control observables a good agreement between the yields of selected events in data and simulation can be observed when neglecting inclusive normalization effects. For the azimuth angle of the hardest jet ϕ^{jet1} the effect of the jet veto (see section 5.1.2) is especially noticeable in the distribution for the data-taking periods 2017 and 2018 depicted in fig. 5.19 significantly reducing the event yields in the affected regions. However, the effect is modelled well in simulation. A normalization of the simulated yields by a constant factor smaller one in all bins and the same for all observables would improve the match. This indicates that the cross section or the luminosity used for the reweighting of the simulated events is overestimated by 5 to 10%. Additionally, the normalization factor slightly differs for each data-taking period. Since the cross section normalization is the same for each period this suggests that the luminosity is separately misestimated. The observed shift in the yields is at the edge but within the uncertainty band including the full treatment of statistical and systematic effects as described in sections 5.5 and 5.6.2.

Combination of Data-Taking Periods – Since there is no significant deviation in the description of the measured event yields by the simulation between the different data-taking periods, the selected yields of each period are combined into a single set by stacking the individual yields following the statistics of weighted Poisson events. From the statistics of weighted Poisson events also the statistical uncertainties are derived (see section 5.6.1). The estimation of the uncertainties related to systematic effects is described separately in section 5.6.2.

Data-Simulation Comparisons in y_b-y^* Phase Space – As an additional control, the yields of selected events in the combined dataset, labelled as *Run 2*, including the full statistics of the recorded data by the CMS collaboration for the proton-proton collisions at 13 TeV at the LHC are compared differentially in the same observables as above but separated in y_b-y^*-bins.

The resulting event yields are shown differentially in

- the pseudorapidity of the two selected oppositely charged muons in figs. 5.21 and 5.22,
- the azimuth angle of the two selected oppositely charged muons in figs. 5.23 and 5.24,
- the transverse momentum of the two selected oppositely charged muons in figs. 5.25 and 5.26,
- the rapidity of the dimuon system in fig. 5.27,

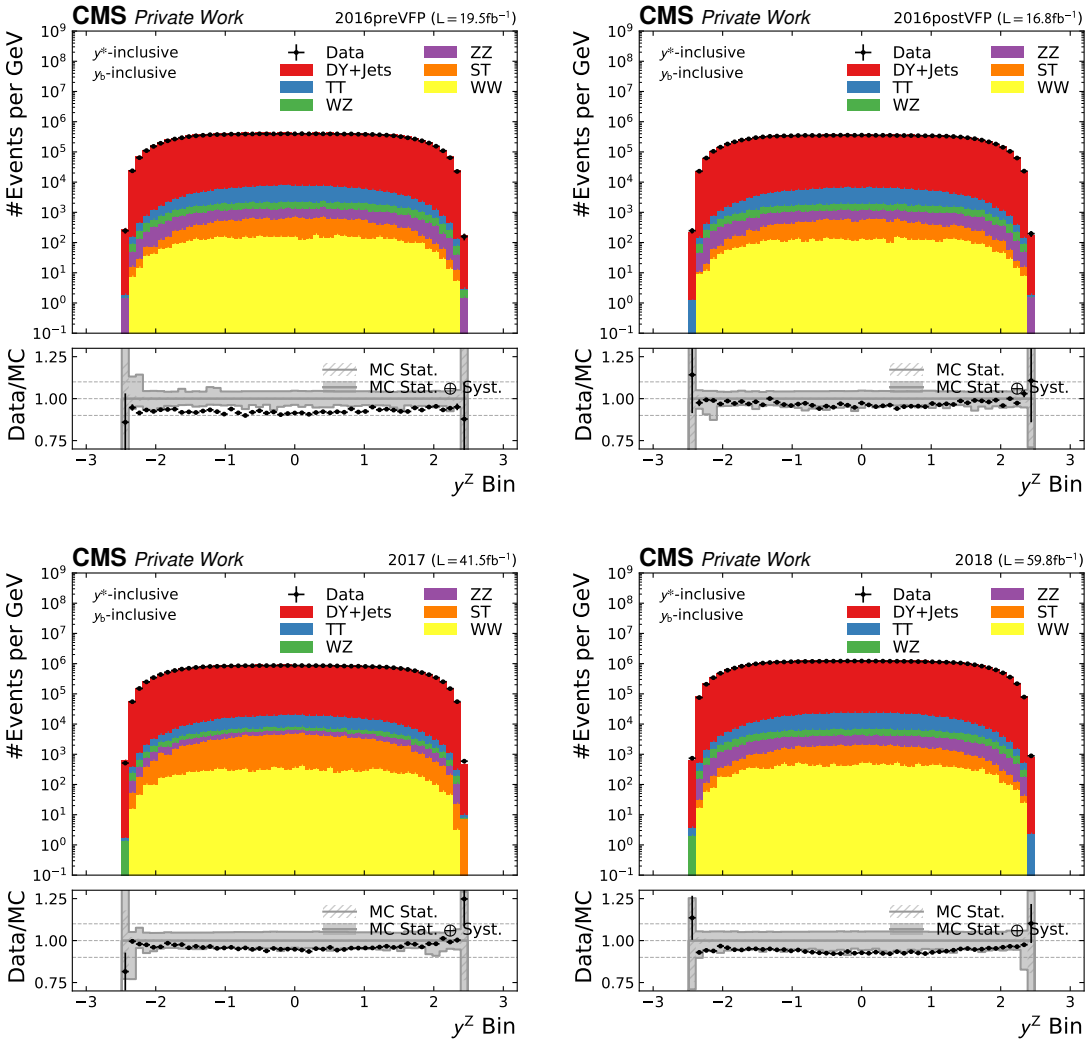


Figure 5.15: Comparison of the rapidity of the dimuon system y^Z at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

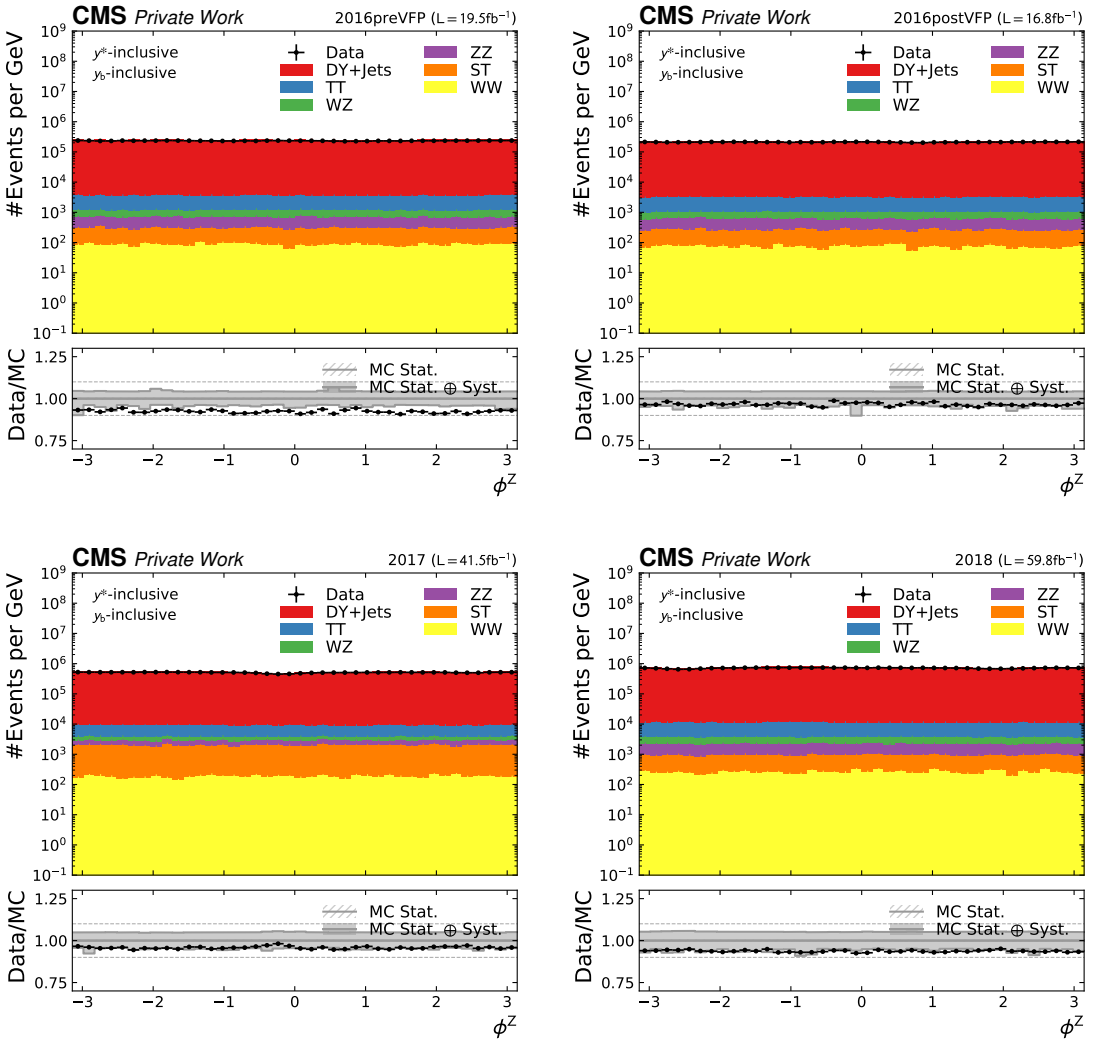


Figure 5.16: Comparison of the azimuth angle of the dimuon system ϕ^Z at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

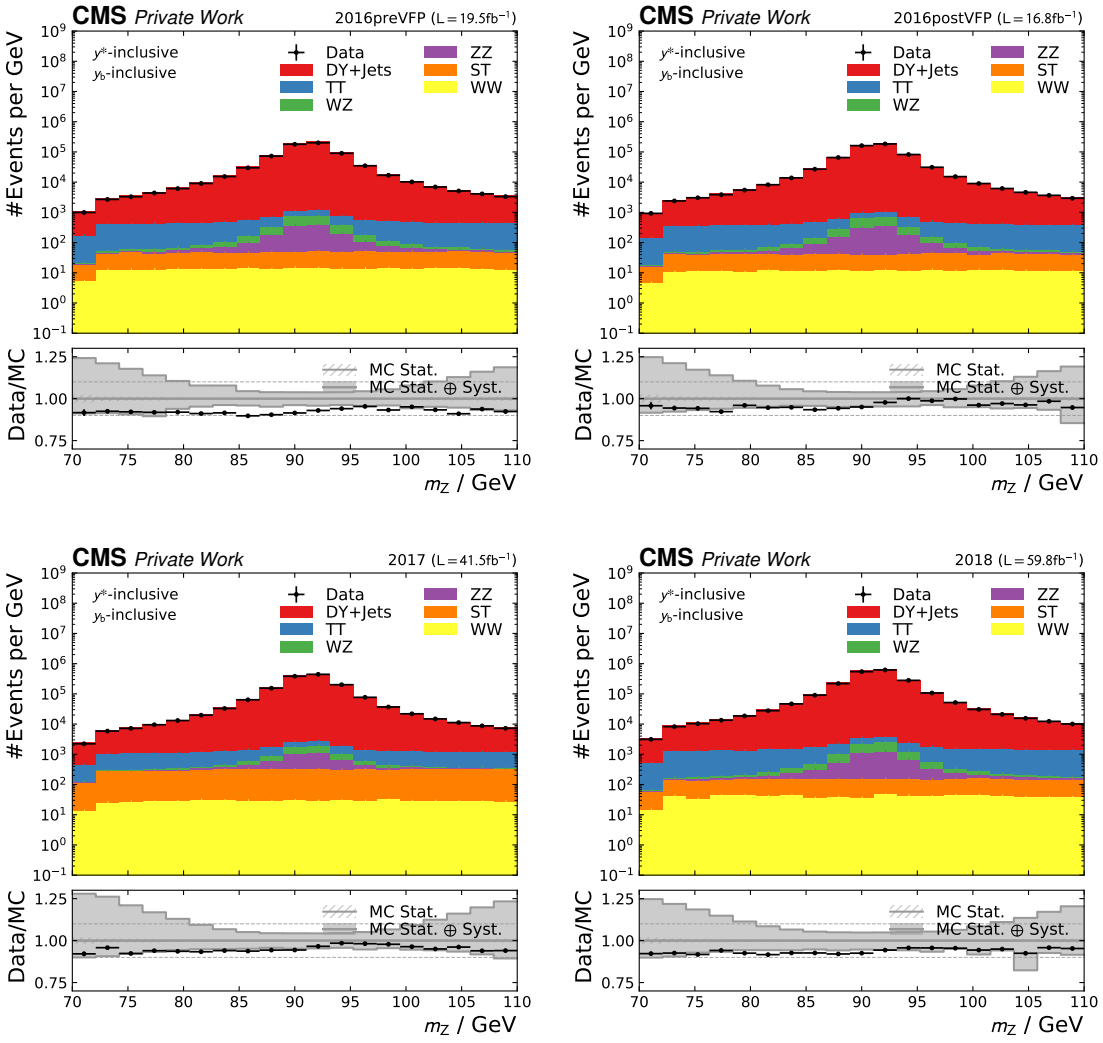


Figure 5.17: Comparison of the invariant mass of the dimuon system m_Z at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

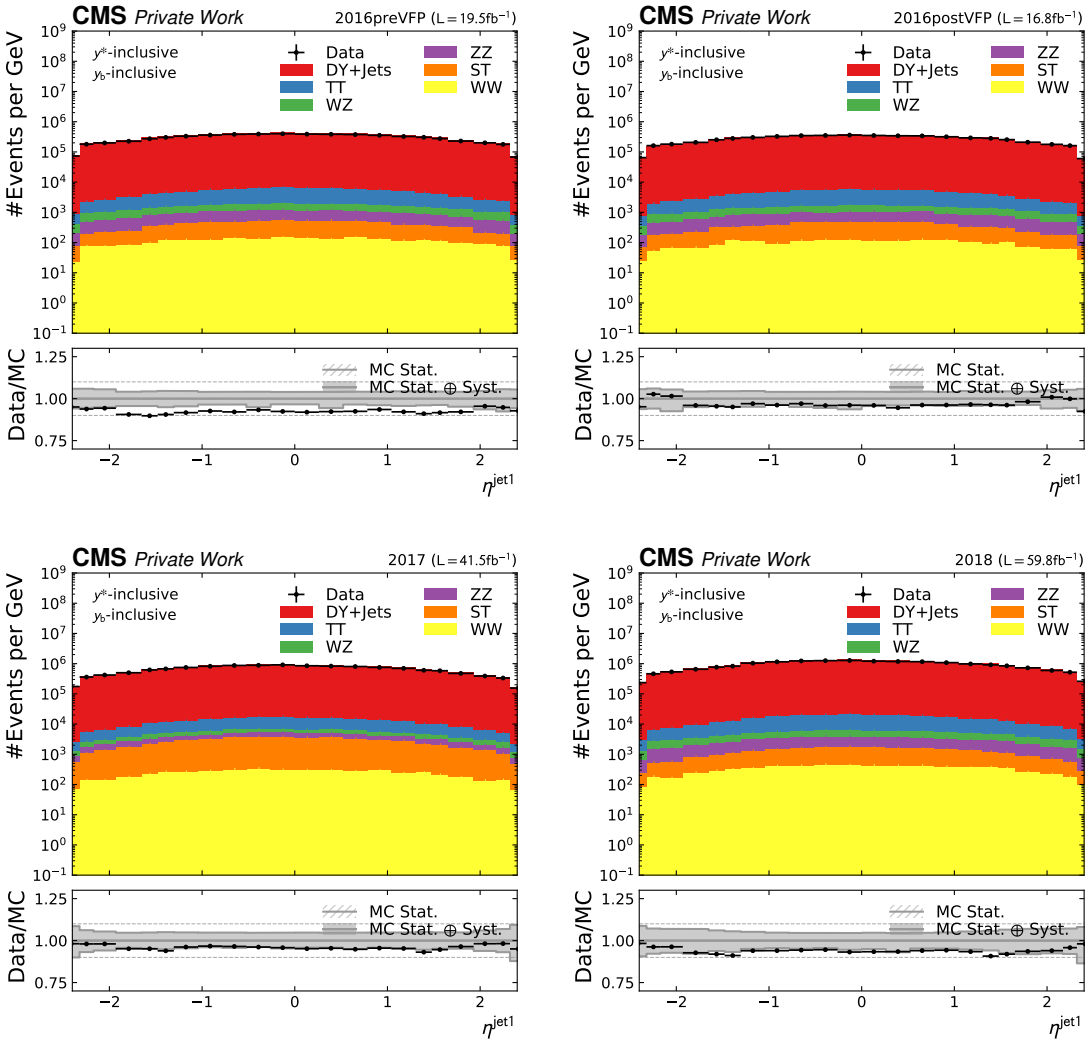


Figure 5.18: Comparison of the pseudorapidity of the hardest jet η^{jet1} at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

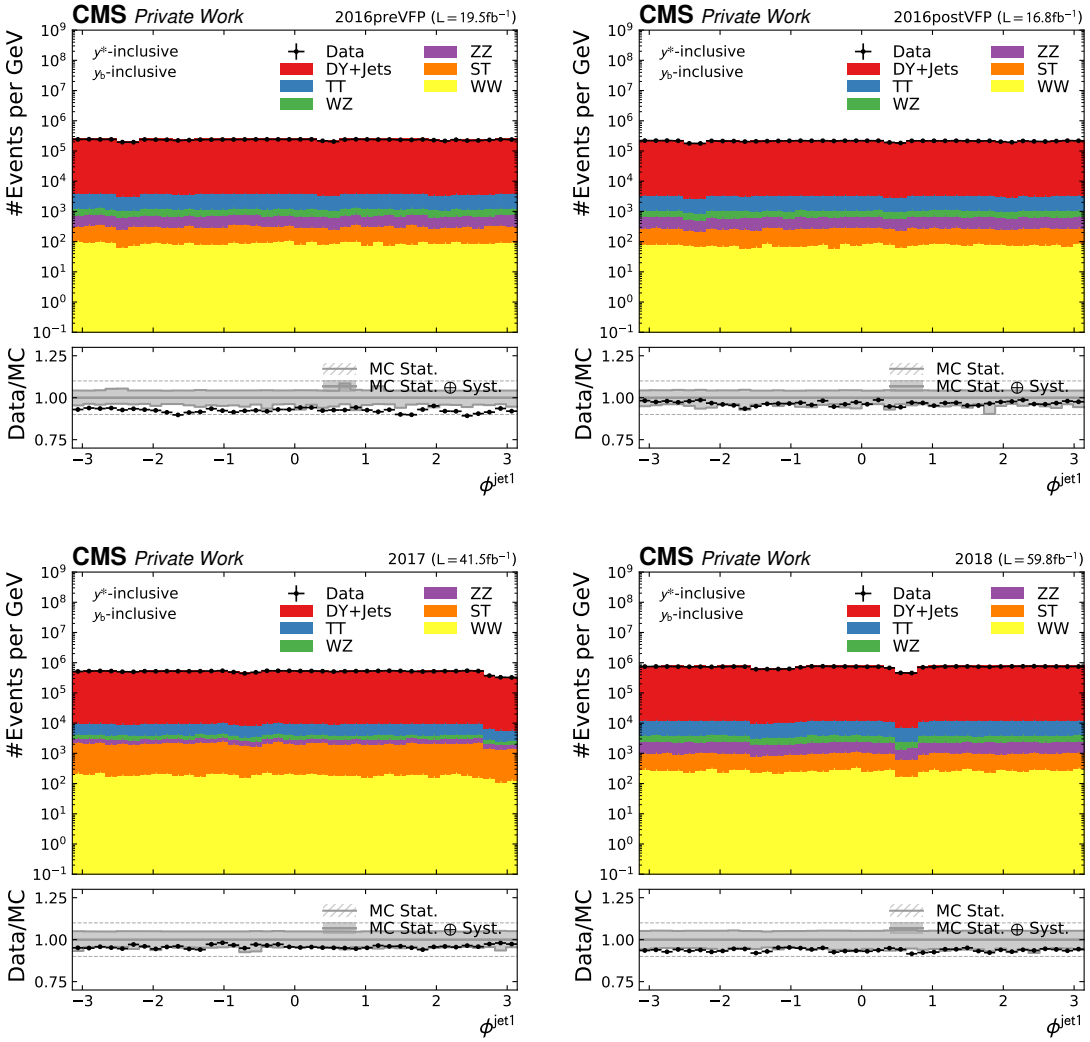


Figure 5.19: Comparison of the azimuth angle of the hardest jet ϕ^{jet1} at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*. The effect of the jet veto (see section 5.1.2) is especially noticeable for high ϕ^{jet1} and $\phi^{\text{jet1}} \sim 0.6$ for the 2017 and 2018 data-taking periods, respectively.

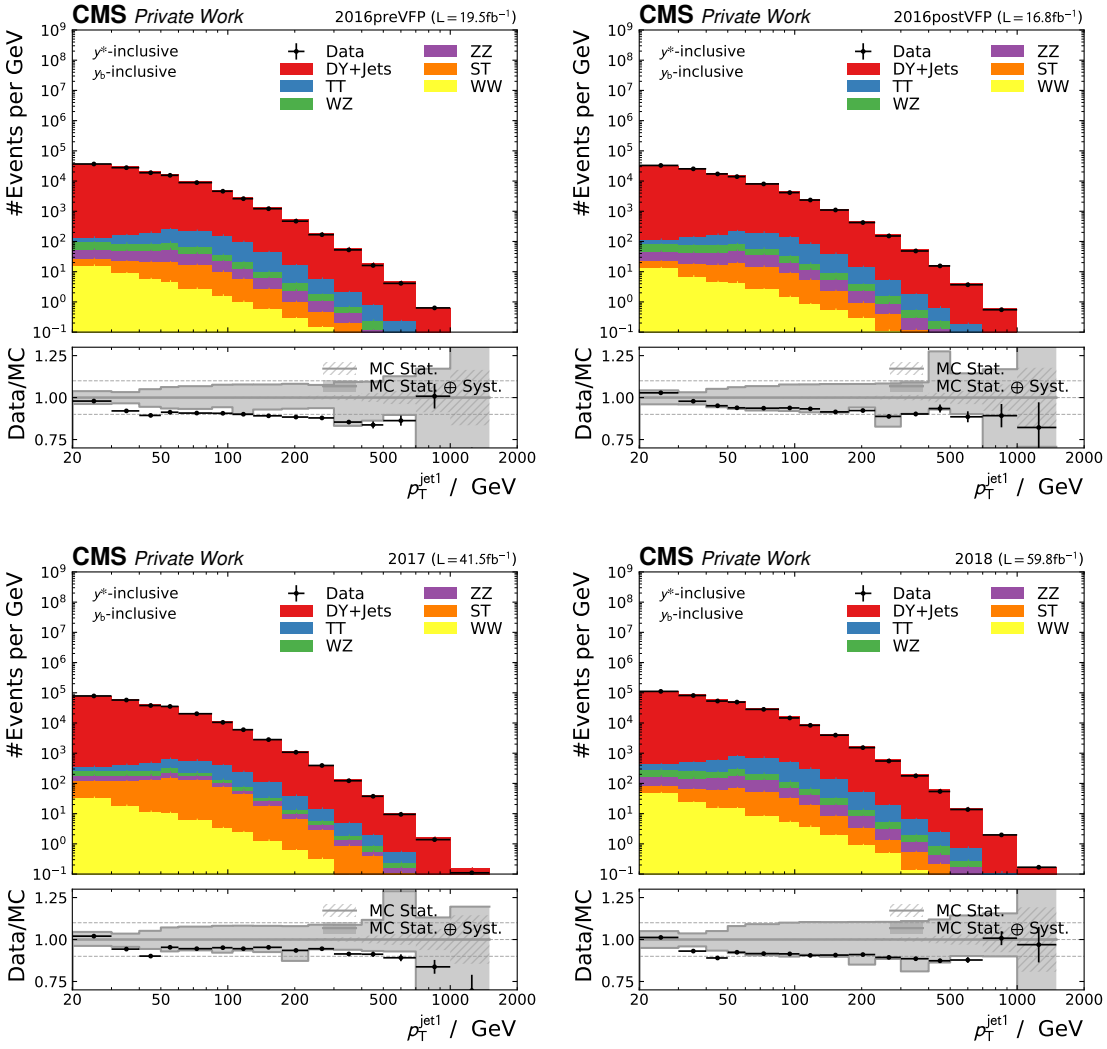


Figure 5.20: Comparison of the transverse momentum of the hardest jet p_T^{jet1} at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*.

- the azimuth angle of the dimuon system in fig. 5.28,
- the invariant mass of the dimuon system in fig. 5.29,
- the pseudorapidity of the hardest jet in fig. 5.30,
- the azimuth angle of the hardest jet in fig. 5.31,
- the transverse momentum of the hardest jet in fig. 5.32

individually for the central y_b-y^*-bin and the two bins with the highest y_b and y^* respectively. The same yields are shown separately for all y_b-y^*-bins in figs. A.7 to A.18.

The shape of the distributions obtained from data and simulation match very well. The overall mismatch in normalization can still be observed but is just at the edge on the uncertainty band and therefore covered by the uncertainties. However, by the splitting of the distributions in individual y_b-y^*-bins, depicted in appendix A.2, an additional dependency of the normalization on y^* is perceived. With increasing y^* the necessary correction factor for accounting for the mismatch in normalization increases indicating an inaccuracy in the modelling. There is no significant dependency on y_b observed. The cause behind this phenomenon remains unclear rendering further research necessary. Nevertheless, as this observation is covered by the assigned statistical and systematic uncertainties, it allows for continued analysis.

Cross-Checks of Final Inputs for Unfolding – The event yields in data are compared to the simulated ones inclusive in y_b-y^* but separated in data-taking periods in fig. 5.33 and differentially in y_b-y^*-p_T^Z bins for the combined dataset in fig. 5.34 for the central y_b-y^*-bin and the two bins with maximum y_b and y^*. They are shown in fig. A.19 for all y_b-y^*-p_T^Z-bins.

In fig. 5.33, the same observations of a difference between the simulated and data yields by an inclusive normalization factor can be observed as in the other observables. Also, the same observed time dependence of that normalization factor is perceived.

In fig. 5.34, the same additional dependence of the normalization factor on y^* is observed as in the other observables. Also here, the observed systematic shifts are covered by the uncertainties in all bins. These observations are compatible with the ones observed in the control observables and no significant deviations are found. This allows for the continued measurement of the differential cross sections using the observed yields in data as an input to the unfolding procedure described in section 5.5.

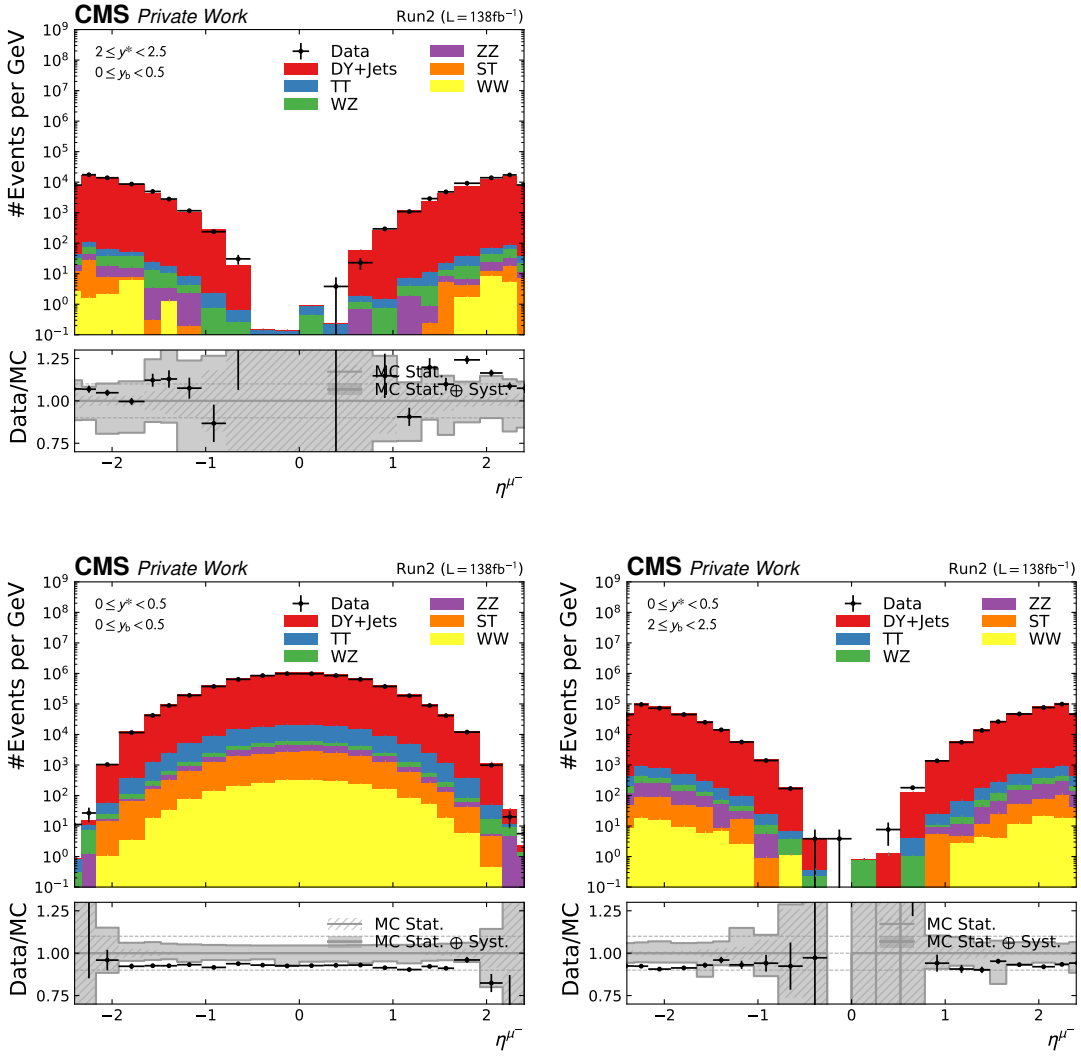


Figure 5.21: Comparison of the pseudorapidity of the positively charged muon selected for the dimuon system reconstruction η^{μ^-} for the combined dataset for the central and two extreme y_b-y^*-bins.

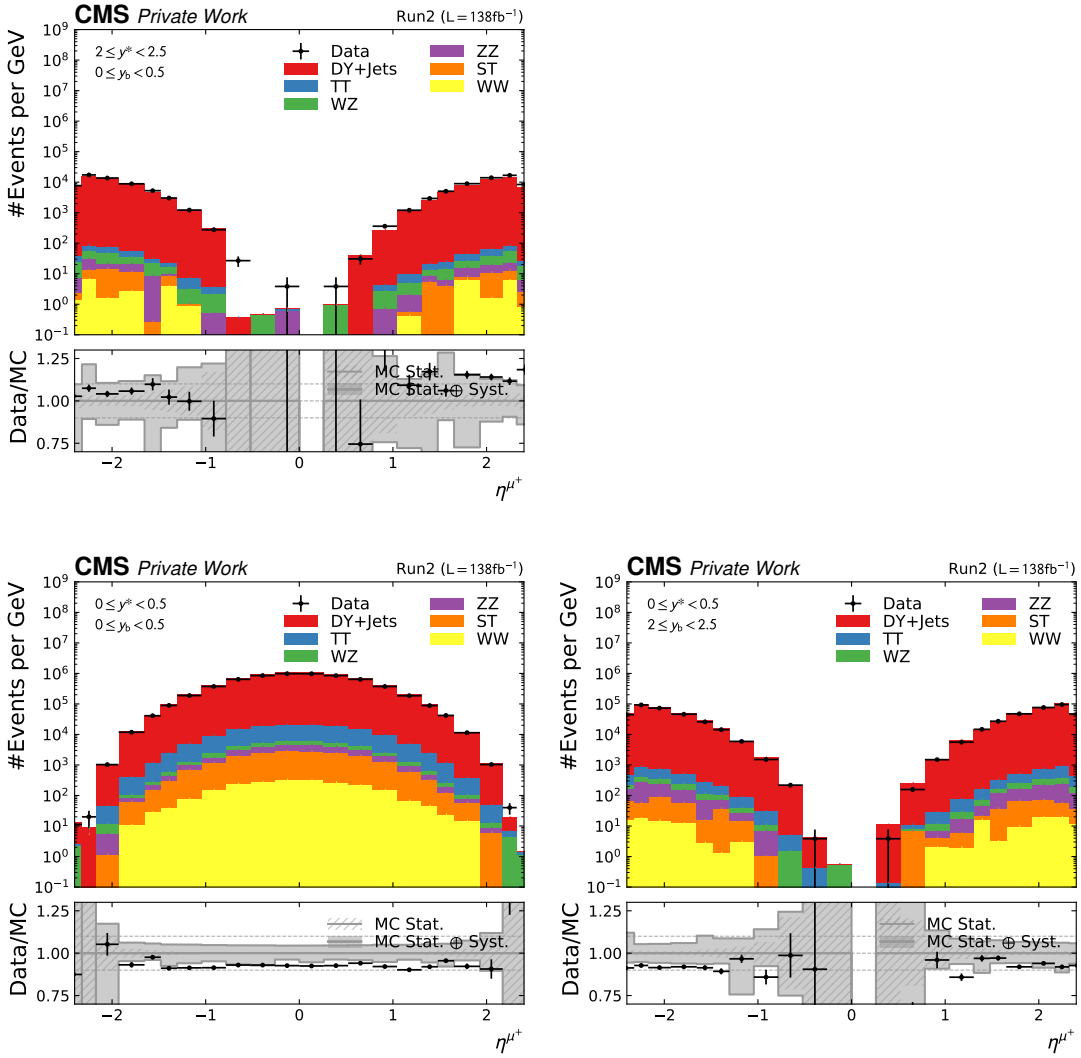


Figure 5.22: Comparison of the pseudorapidity of the positively charged muon selected for the dimuon system reconstruction η^{μ^+} for the combined dataset for the central and two extreme y_b-y^*-bins.

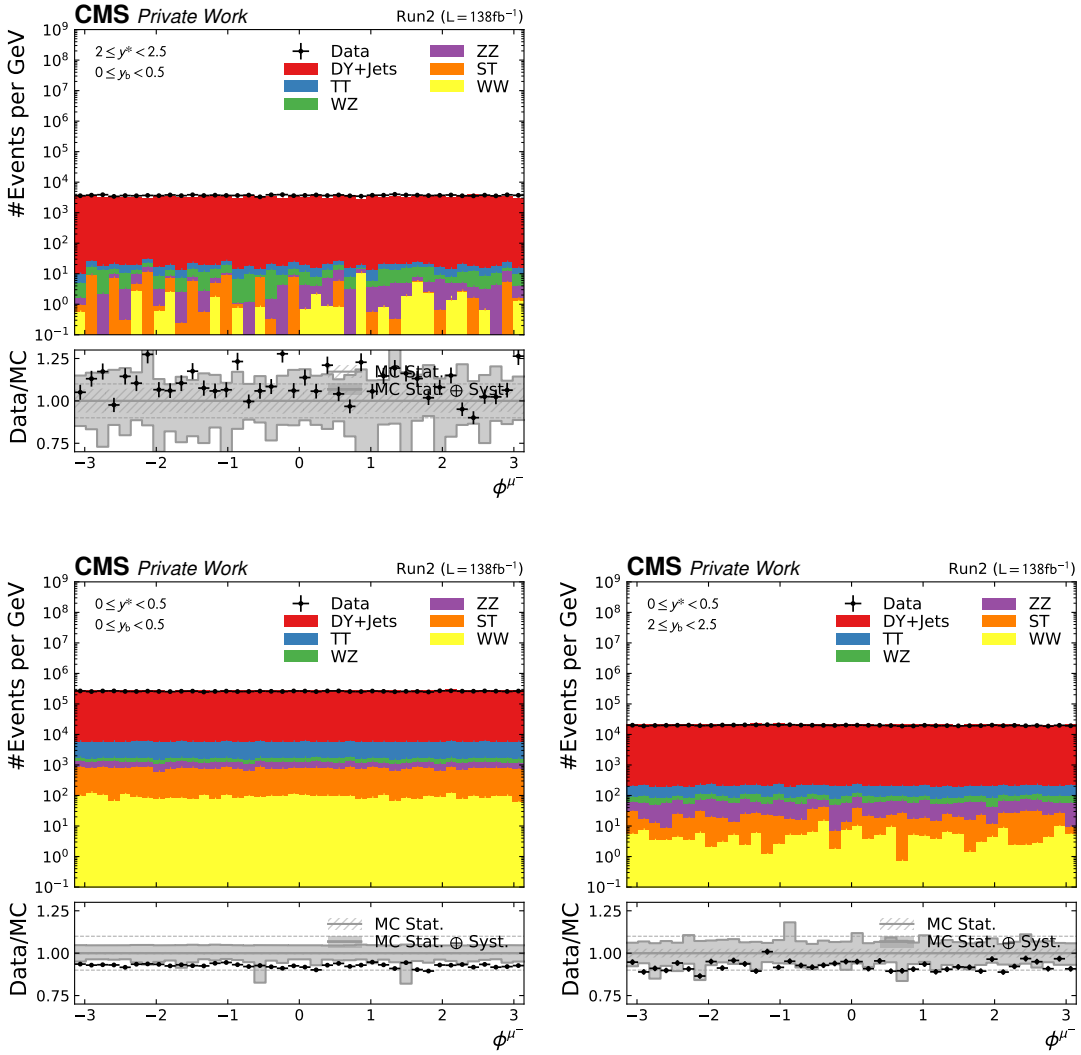


Figure 5.23: Comparison of the azimuth angle of the positively charged muon selected for the dimuon system reconstruction ϕ^{μ^-} for the combined dataset for the central and two extreme y_b-y^*-bins.

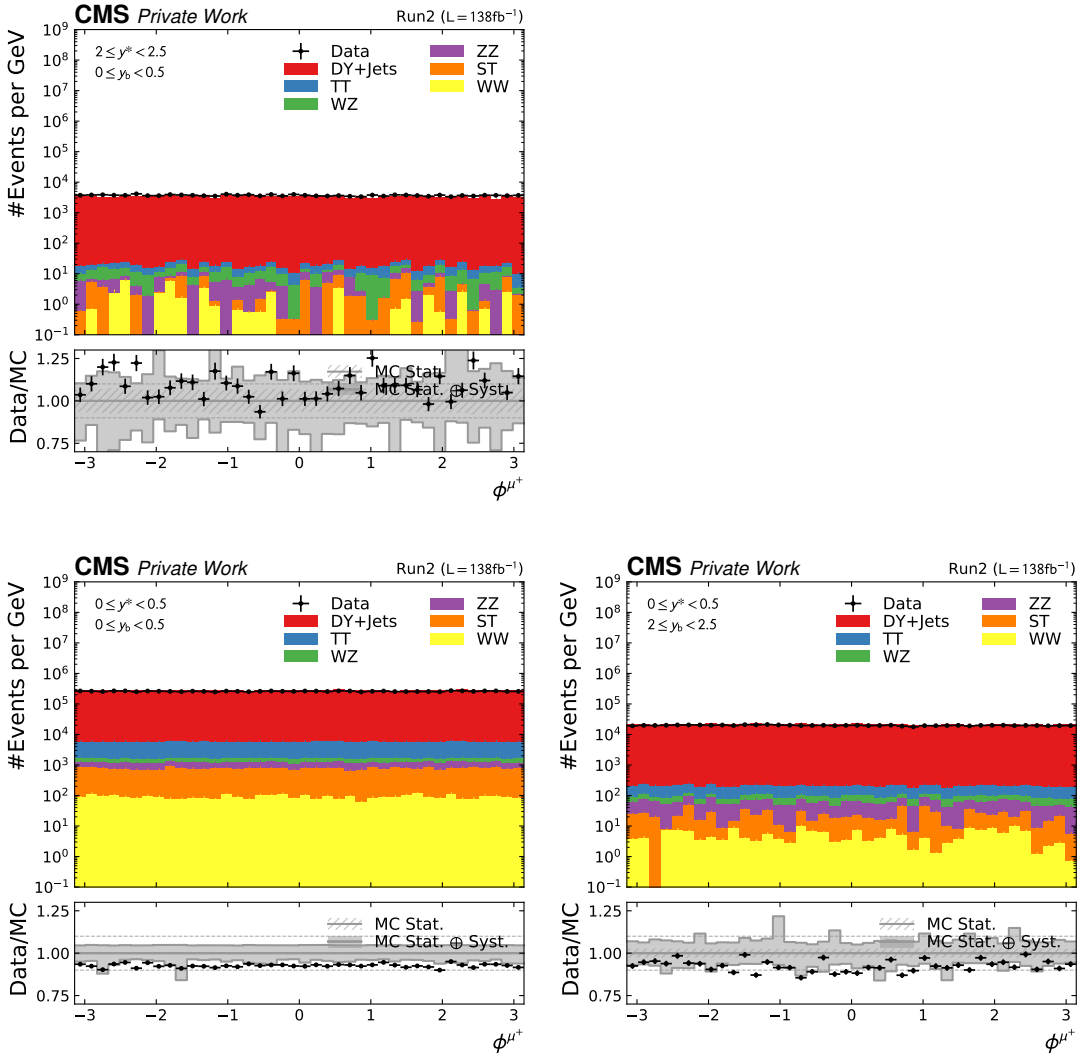


Figure 5.24: Comparison of the azimuth angle of the positively charged muon selected for the dimuon system reconstruction ϕ^{μ^+} for the combined dataset for the central and two extreme y_b-y^*-bins.

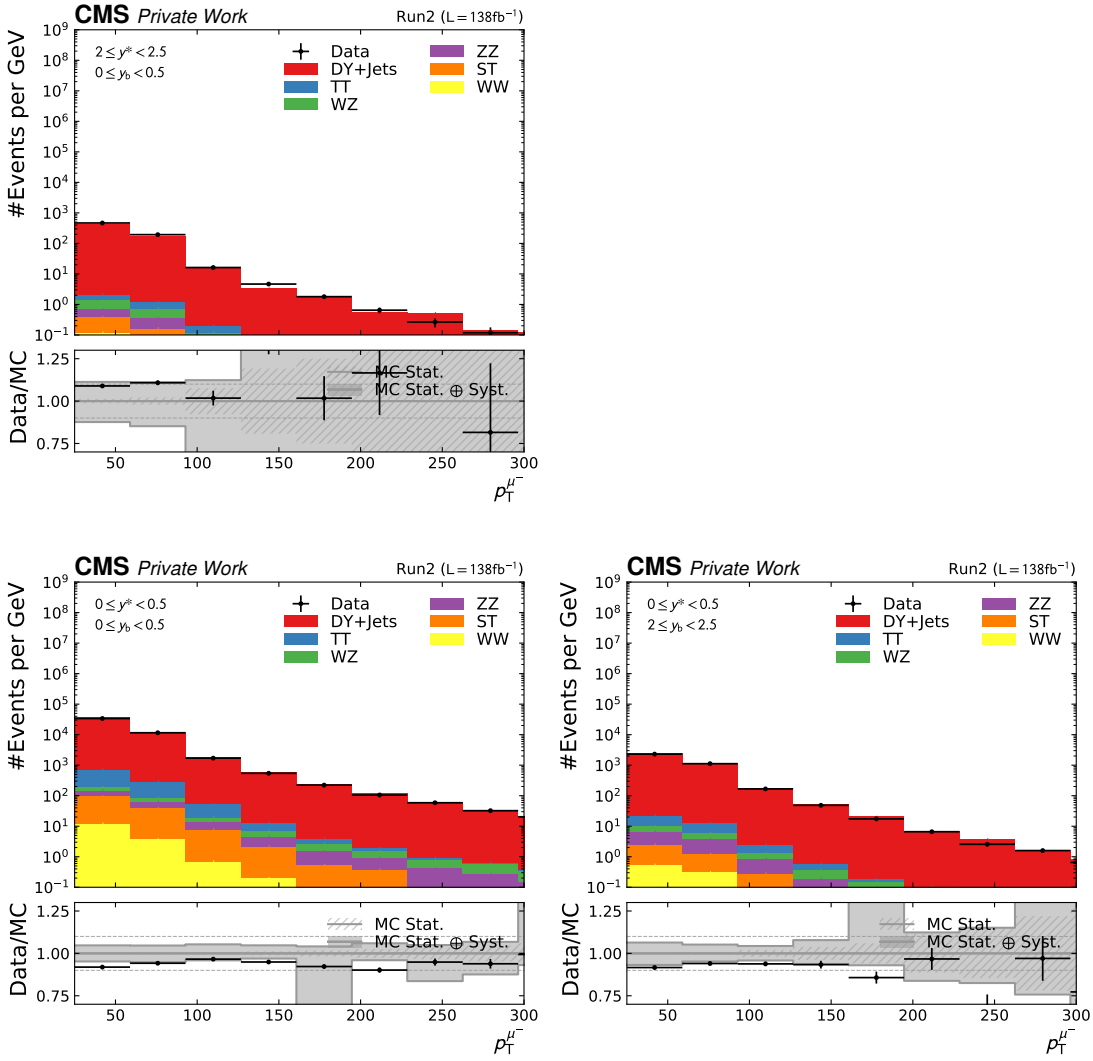


Figure 5.25: Comparison of the transverse momentum of the positively charged muon selected for the dimuon system reconstruction $p_T^{\mu^-}$ for the combined dataset for the central and two extreme y_b-y^*-bins.

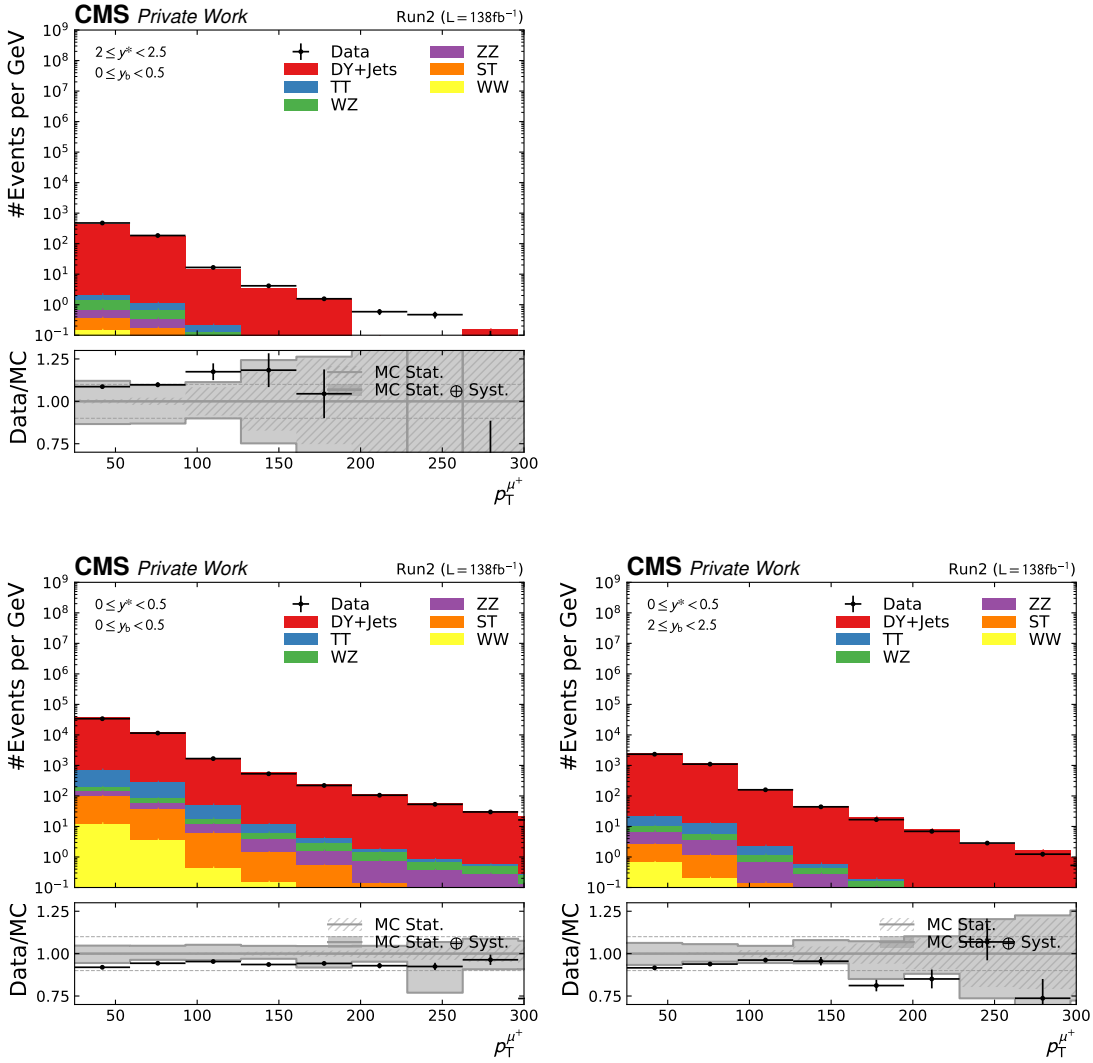


Figure 5.26: Comparison of the transverse momentum of the positively charged muon selected for the dimuon system reconstruction $p_T^{\mu^+}$ for the combined dataset for the central and two extreme y_b-y^*-bins.

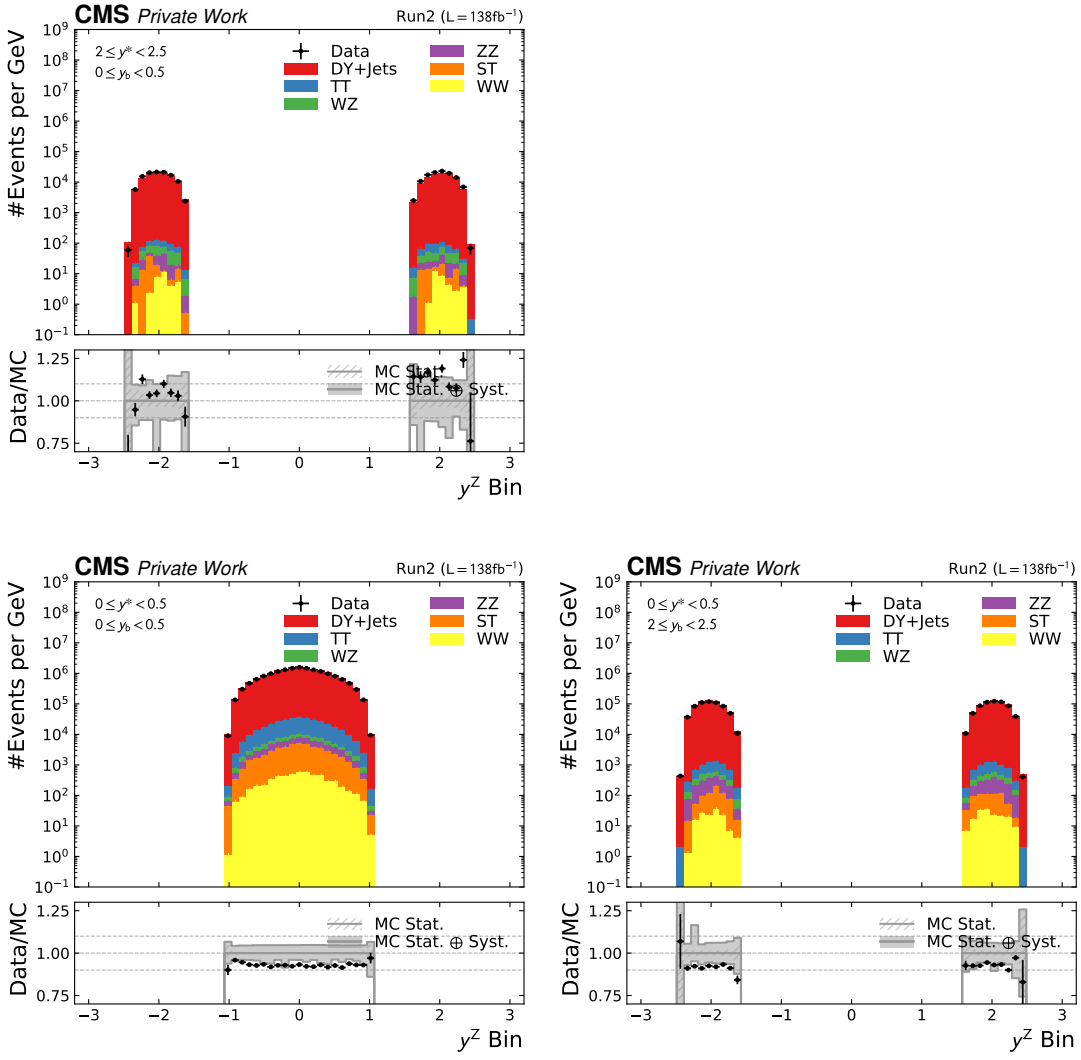


Figure 5.27: Comparison of the rapidity of the dimuon system y^Z for the combined dataset for the central and two extreme y_b-y^*-bins.

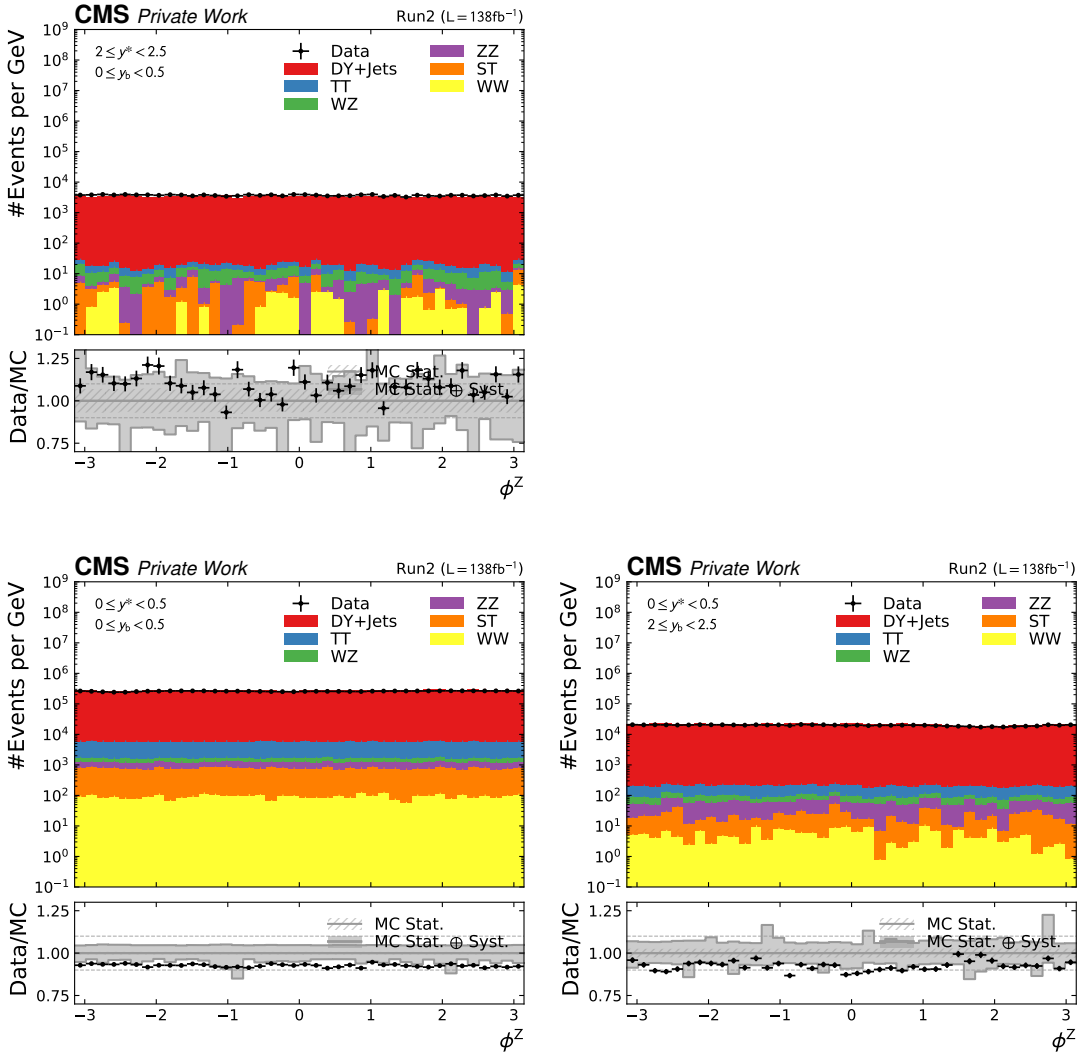


Figure 5.28: Comparison of the azimuth angle of the dimuon system ϕ^Z for the combined dataset for the central and two extreme y_b-y^*-bins.

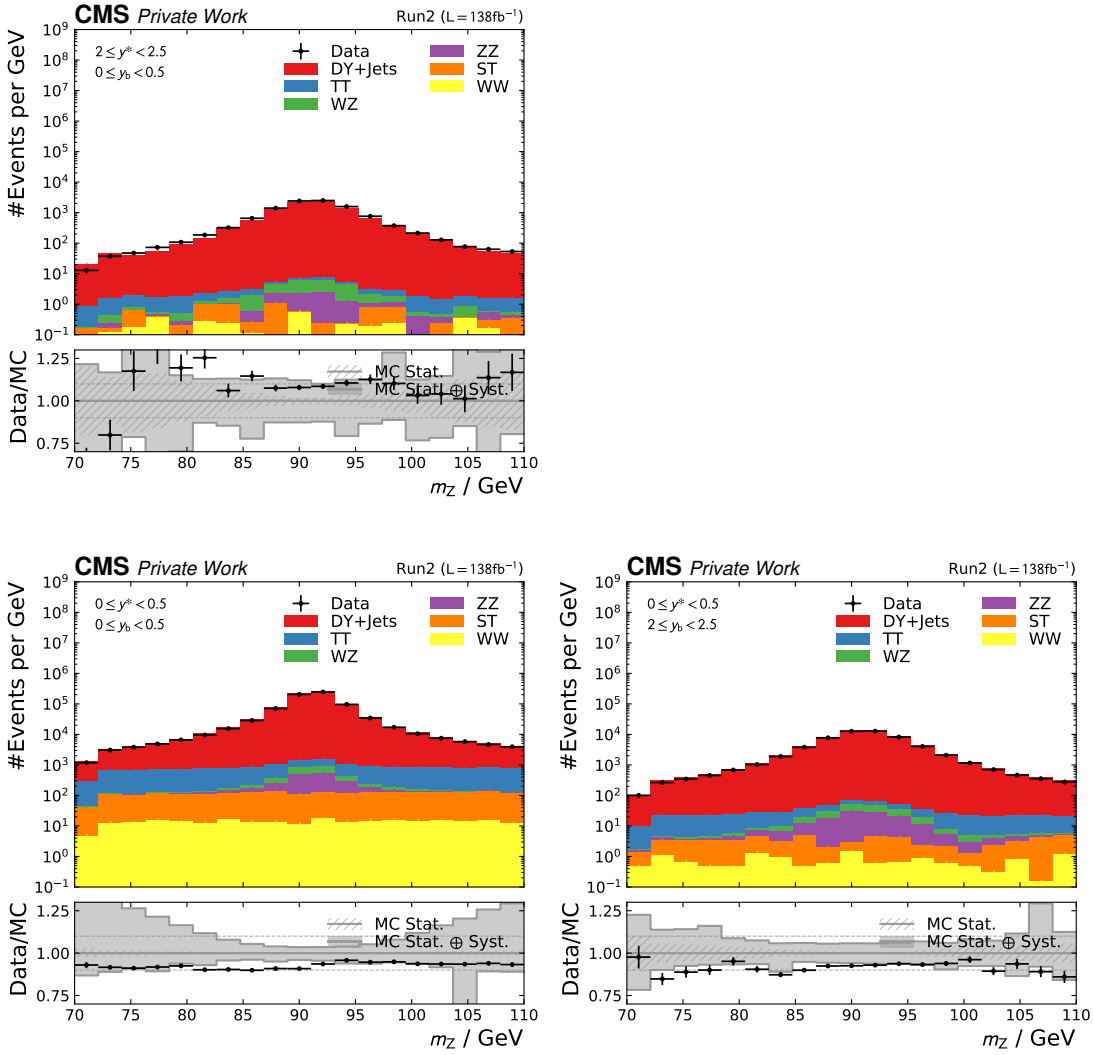


Figure 5.29: Comparison of the invariant mass of the dimuon system m_Z for the central and two extreme y_b-y^*-bins for the combined dataset.

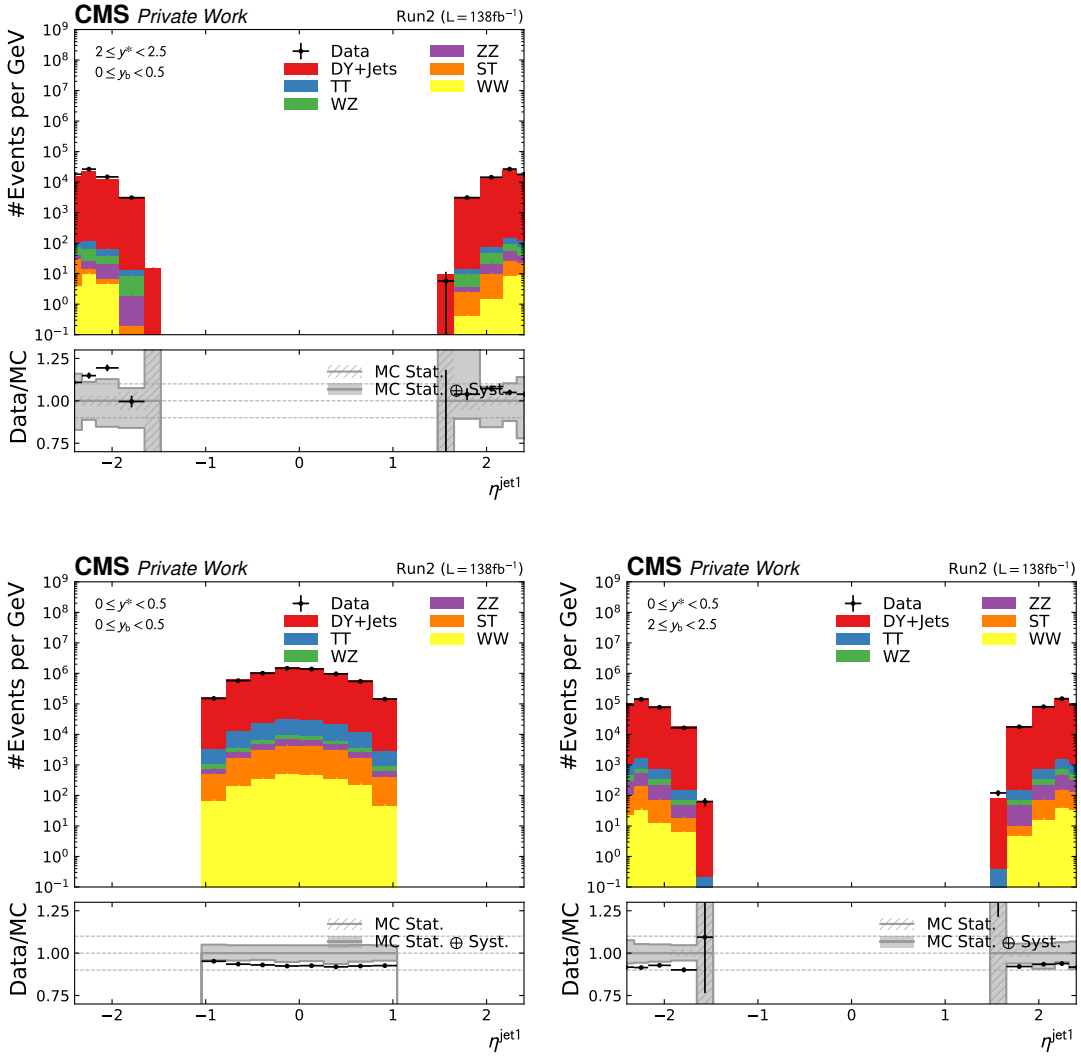


Figure 5.30: Comparison of the pseudorapidity of the hardest jet η^{jet1} for the central and two extreme y_b-y^*-bin for the combined dataset.

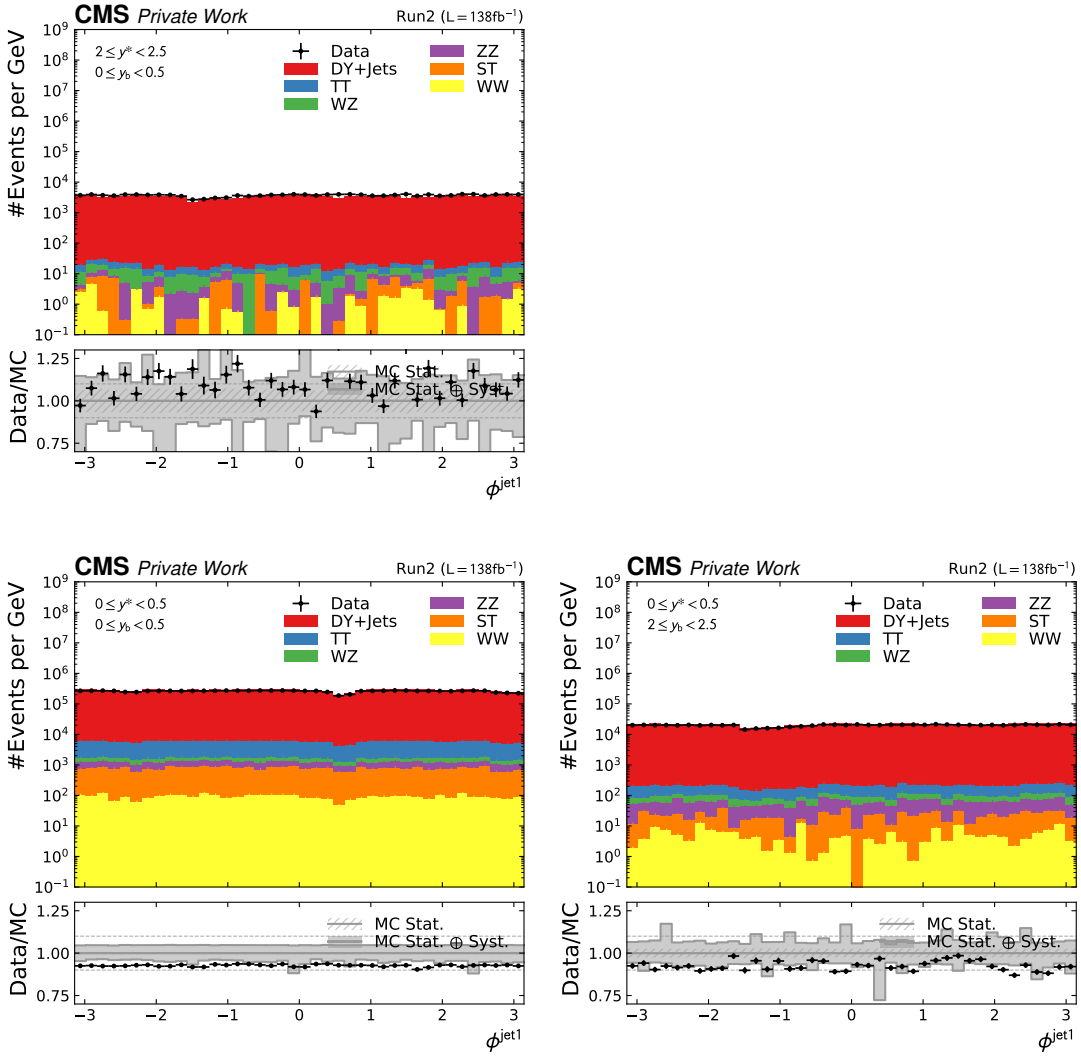


Figure 5.31: Comparison of the azimuth angle of the hardest jet ϕ^{jet1} for the central and two extreme y_b-y^*-bin for the combined dataset.

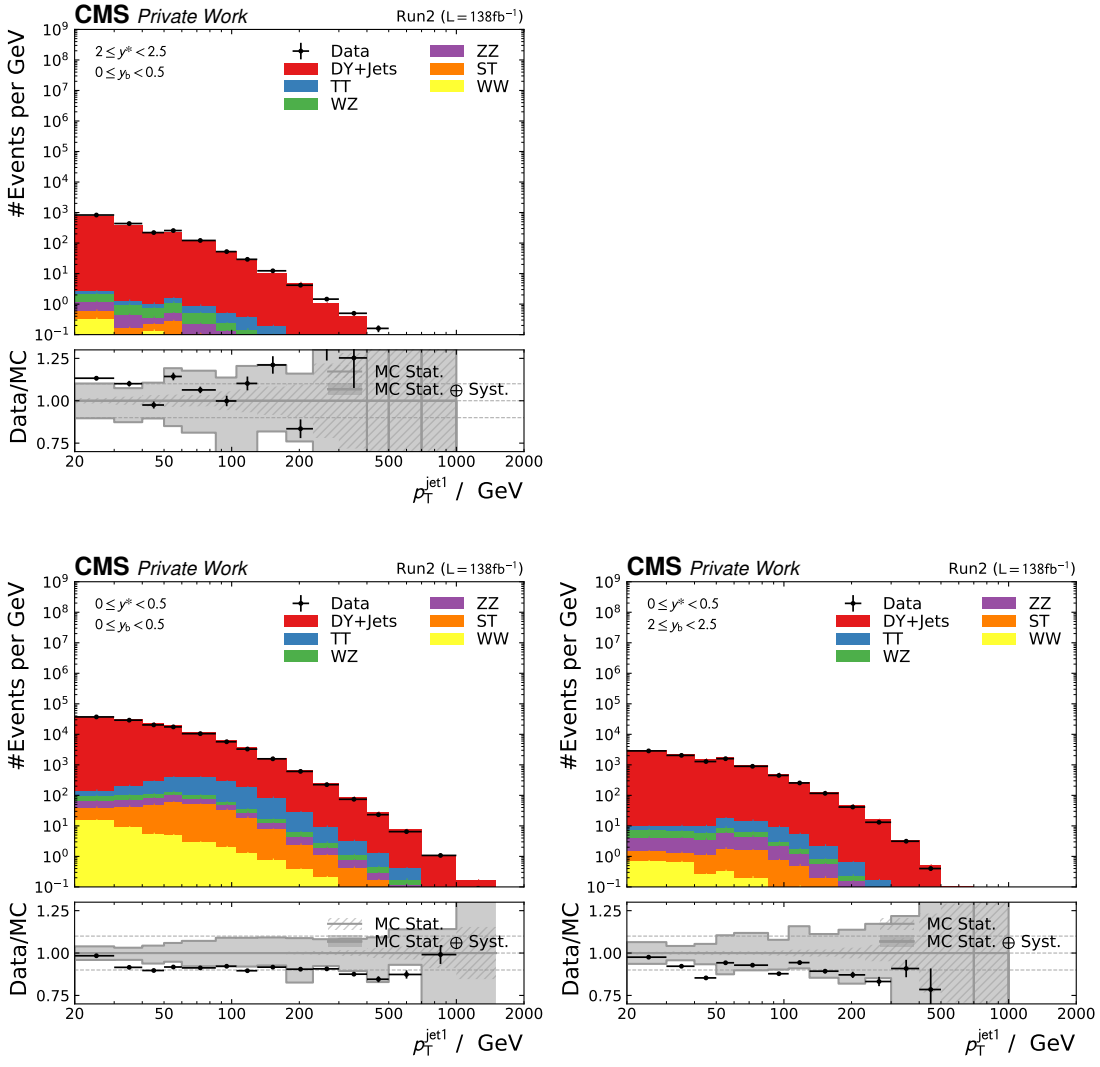


Figure 5.32: Comparison of the transverse momentum of the hardest jet p_T^{jet1} for the central and two extreme y_b-y^*-bins for the combined dataset.

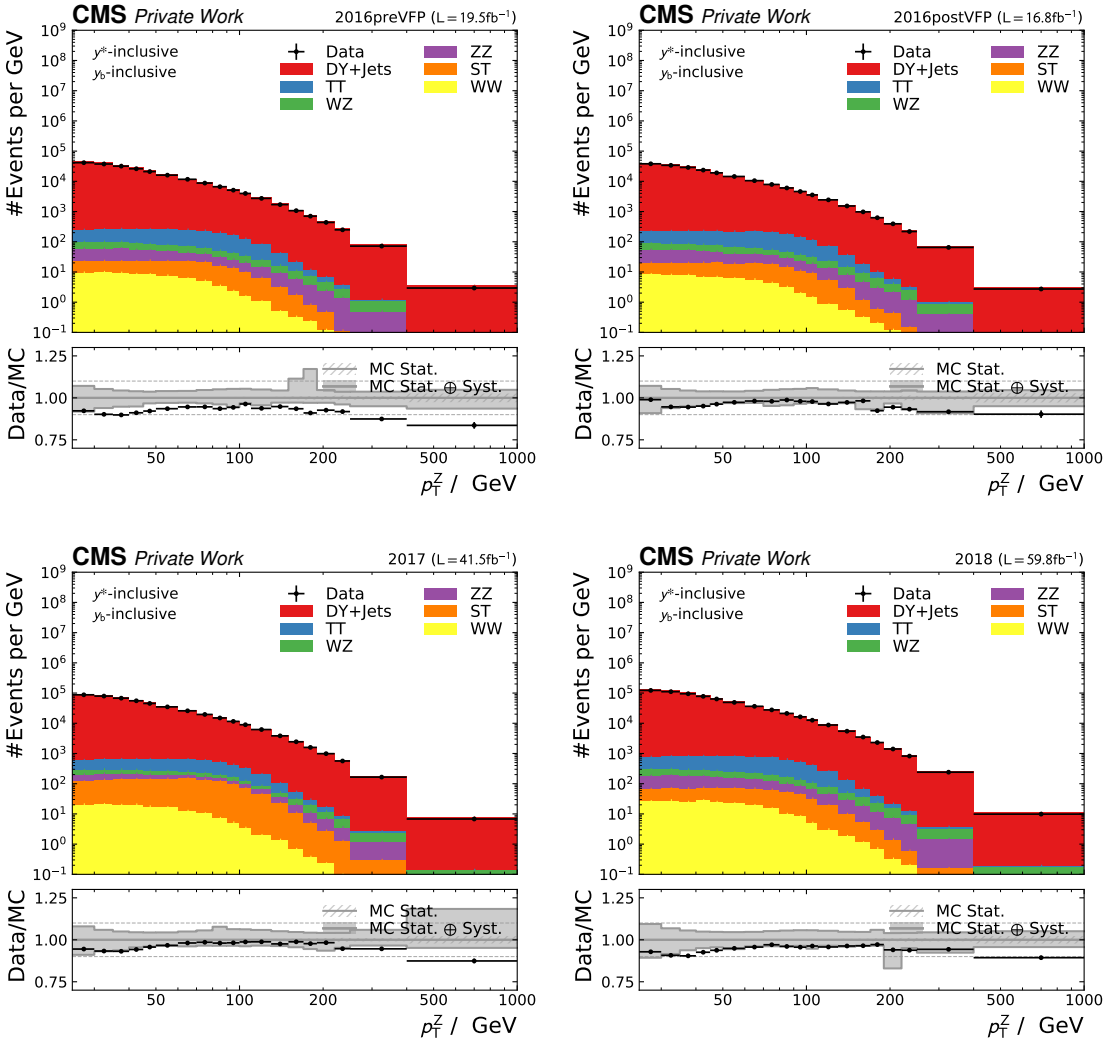


Figure 5.33: Comparison of the transverse momentum of the dimuon system p_T^Z at reconstruction level for each of the four data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018 inclusive in y_b-y^*. The observed differential shapes of the event yields predicted by the stacked signal and background simulations agree with the ones selected in data within uncertainties. A systematic bias for an inclusive normalization factor of the simulation is indicated by a shift of approximately 2 to 10% with respect to data. The normalization factor differs slightly between individual data-taking periods.

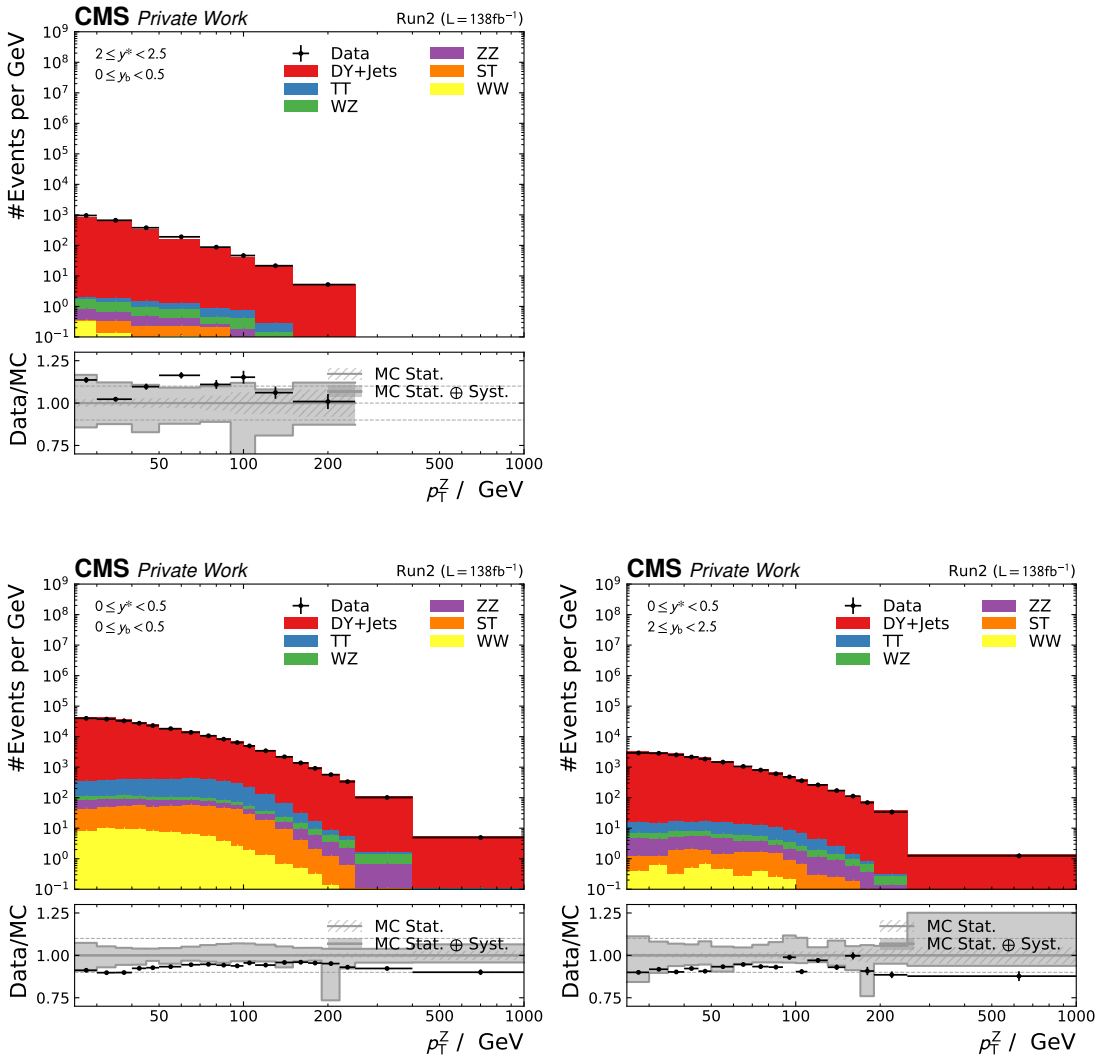


Figure 5.34: Comparison of the transverse momentum of the dimuon system p_T^Z at reconstruction level for the central and two extreme y_b-y^*-bins for the combined dataset. The observed differential shapes of the event yields predicted by the stacked signal and background simulations agree with the ones selected in data within uncertainties. A systematic bias dependent on the y_b-y^*-bin is indicated by a shift of the simulation with respect to data. The normalization factor grows from approximately 95% to 110% with increasing y^*. No dependence on y_b is observed.

5.5 Mitigation of Detector Effects and Derivation of the Cross Sections

To derive the cross sections, the event yields obtained differentially in y_b, y^* and p_T^Z from the combined dataset of the fully corrected and quality assessed selected events need to be corrected for acceptance and efficiency effects. These effects are introduced by selection criteria, the fiducial phase space coverage and dead zones of the detector, and limited efficiencies of the reconstruction of the analysed objects and corrections. Furthermore, migration of events among the analysed phase space bins occur due to the limited resolution of the detector. The migration can be only assessed as random effects due to the probabilistic nature of the interactions of particles with the detector components and the complexity of the reconstruction. This needs to be accounted for in the derivation of the differential cross sections.

The mitigation of both described effects caused by the detector is performed simultaneously using a so-called unfolding procedure described in the following. It derives the cross sections differentially in the y_b-y^*-p_T^Z bins corrected for the detector effects allowing a direct comparison with theoretical predictions without a simulation of the detector.

5.5.1 Unfolding Procedure

In general, the idea of unfolding is to reverse the inevitable effects of the detector on a true observable or set of observables $t(x)$ dependent on the true phase space coordinates x. The measurement of an reconstructed observable or set of reconstructed observables $s(y)$ dependent on reconstructed phase space coordinates y is given as the folding integral

$$s(y) = \int_X D(y,x)t(x)\mathrm{d}x \tag{5.17}$$

with the integration performed over the true phase space X with coordinates x. The effects of the limited detector resolution are encoded in the folding function $D(y,x)$. These effects lead to a change in the population of events in the reconstructed phase space coordinates compared to the true phase space. This change is due to migrations, and the acceptance and efficiency limitations leading to a loss of events in the reconstructed phase space. To reverse the effect of $D(y,x)$ an inverse transformation $D'^{-1}(x,y)$ is needed which gives access to the true observables in the true phase space

$$t(x) = \int_Y D'^{-1}(x,y)s(y)\mathrm{d}y \tag{5.18}$$

with integration over the reconstructed phase space Y with coordinates y.

In practice, the observables are measured in a discretized phase space. Consequently, eq. (5.17) transforms to

$$s^i = \sum_j \mathbf{R}_j^i t_j \tag{5.19}$$

with vectors of the reconstructed observables s in bins i and true observables t in bins j. The migration matrix

$$\mathbf{R}_j^i = \frac{\int_{Y_i}\int_{X^j} D(y,x)t(x)\mathrm{d}x\mathrm{d}y}{\int_{X^j} t(x)\mathrm{d}x} \tag{5.20}$$

with integrations within the bin boundaries of the respective bin on the true phase space X^j and reconstructed phase space Y_i encodes the migrations and event losses from the true to the reconstructed binned observables. Given the migration matrix eq. (5.18) simplifies to inverting $\mathbf{R}_j^i$ and applying it to the measured reconstructed observables leading to the true observables

$$t_j = \mathbf{R^{-1}}{}_j^i s^i \tag{5.21}$$

with $\mathbf{R}_j^i \mathbf{R^{-1}}{}_j^i = \mathbb{1}$. Finding this inverse transformation and applying it onto the measured reconstructed observables s is called *unfolding*. Numerical and algebraic unfolding methods widely used in HEP are for instance the D'Agostini method [130] or TUnfold [131]. The latter is chosen for unfolding in this analysis.

Regularization – Unfolding poses however an ill-posed inverse problem since the reconstructed observables as well as the true observables are subject to statistical fluctuations due to their probabilistic nature. When $\mathbf{R^{-1}}{}_j^i$ is sensitive to perturbations small perturbations in the input have big effects on the solution rendering the procedure unreliable. In that case the inverse problem needs to be regularized. Multiple regularisation methods for unfolding purposes for instance Tikhonov regularisation [132] implemented in TUnfold, the regularisation implemented in the D'Agostini method [130], or Singular Value Decomposition [133] exist. An estimate for the necessity of regularisation in a particular problem can be made utilising the condition number [134] which gives a scalar value for how much the output depends on small changes in the input.

Unfolding Method TUnfold – For this analysis, TUnfold is used for the unfolding of the event yields measured in data. The algorithm estimates a true set of observables t from a measured set of observables s assuming Gaussian distributed s with expectation value

$$\hat{s} = \mathbf{R}\hat{t} \tag{5.22}$$

with the migration matrix $\mathbf{R}$ transforming the expectation value at true level $\hat{t}$. With this Gaussian assumption the true level observables can be estimated by maximizing the least-squares likelihood

$$\mathcal{L}(t) = (s - \mathbf{R}t)^T \mathbf{V_{ss}^{-1}} (s - \mathbf{R}t) \tag{5.23}$$

with given s, $\mathbf{R}$, and covariance matrix at reconstruction level $\mathbf{V_{ss}}$.

Equation (5.23) can be analytically solved for the optimal solution $t_0(s, \mathbf{R}, \mathbf{V_{ss}})$ that maximises the likelihood $\mathcal{L}$ and the resulting unfolded covariance matrix on truth level t, $\mathbf{V_{tt}}(\frac{\partial t_0}{\partial s}, \mathbf{R}, \mathbf{V_{ss}})$. Similarly, an analytical solution for additional contributions to $\mathbf{V_{tt}}$ due to uncertainties on $\mathbf{R}$ can be derived. Consequently, when all inputs s, $\mathbf{R}$, and $\mathbf{V_{ss}}$ are known, an algebraic calculation is sufficient to obtain the true observables t and the corresponding statistical uncertainties.

For small deviations of the underlying probability distributions of s and t from a Gaussian, a normalization term $\mathcal{L}_{\text{norm}}$ is added to eq. (5.23). For large deviations, the TUnfold method is unsuitable and a different method is needed for the unfolding. This is not the case here. Instead, the observables can be directly related to a weighted sum of Poisson events (see section 5.4). Furthermore, the number of events associated to each analysed phase space bin is large enough for approximating the Poisson distribution as a Gaussian without needing the additional normalization term.

When the migration matrix R is ill-conditioned a regularisation is needed and a regularisation term $\mathcal{L}_{\text{reg}}$ is added to eq. (5.23) which implements the Tikhonov regularisation method [132]. In this analysis, the migration matrix is well-conditioned and no regularisation is needed.

5.5.2 Unfolding Inputs

The input for the unfolding method is the migration matrix obtained from simulation. Using the migration matrix, the measured event yields in data minus the expected background contributions s in all y_b-y^*-p_T^Z-bins can be unfolded. The corresponding statistical uncertainties are encoded in a diagonal covariance matrix $\mathbf{V_{ss}}$ assuming the event yields in the individual bins to be uncorrelated.

Estimation of Migration Matrix – The migration matrix $\mathbf{R}_j^i$ is estimated using the simulated events for the signal process (see section 5.3). For each of these events the true contribution to the measured differential cross sections is known at generation level. By subsequent simulation their corresponding contribution at reconstruction level is estimated (see section 5.1.3).

Consequently, the migration matrix can be inferred in a two-dimensional representation, one dimension for the 264 indexed y_b-y^*-p_T^Z bins (see section 5.1.1) on reconstruction and generation level each. It is constructed by applying the selection cuts on the corresponding variables at generation level and the fully reconstructed, corrected, and quality assessed variables at reconstruction level (see section 5.1.3) for each event. Based on the selected bins at generation and reconstruction level the indices $0 \leq i \leq 264$ and $0 \leq j \leq 264$ of the respective y_b-y^*-p_T^Z-bins of a two-dimensional histogram representing the first part of the migration matrix are determined. Subsequently, the events' weights with all (efficiency) corrections applied are filled into the corresponding elements of the matrix. After all events have been filled each column in the generation level dimension is

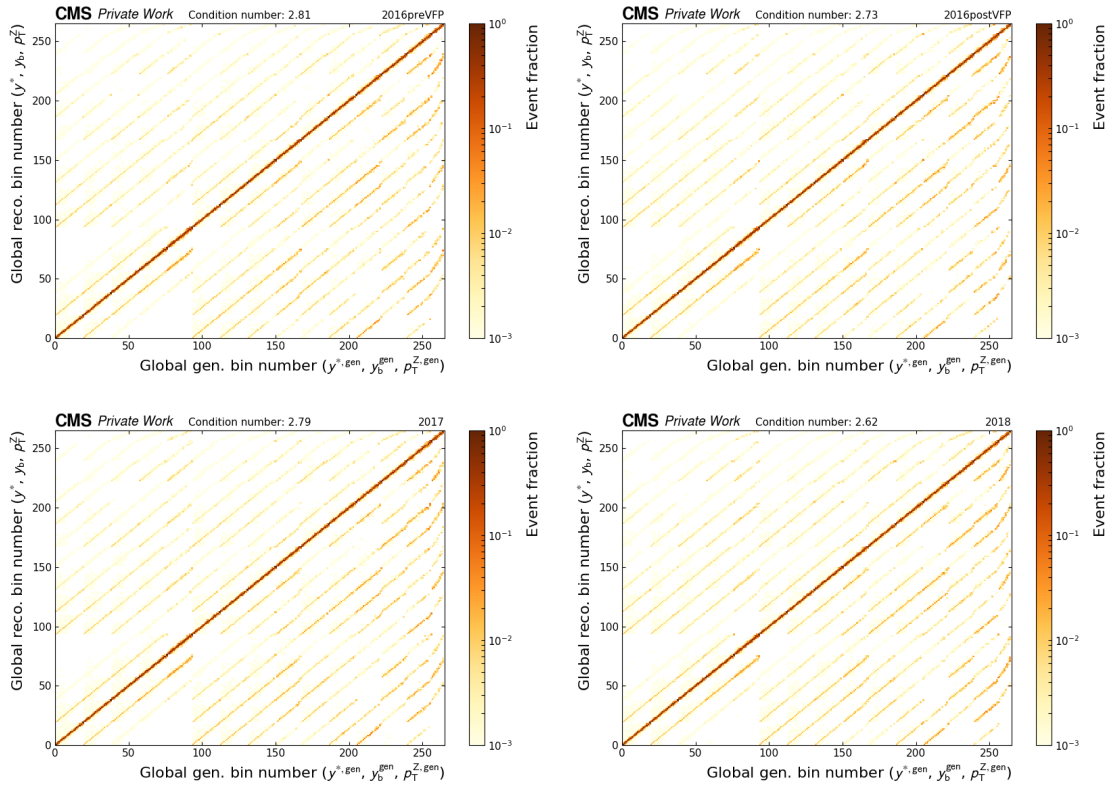


Figure 5.35: Migration matrices constructed (see section 5.5.2) for the individual data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018, (see sections 5.2 and 5.3) respectively. All migration matrices contain mostly entries on the diagonal with small migrations to neighboring bins in phase space; note that due to the one dimensional representation of the phase space chosen in this visualization neighboring bins in y_b and y^* are not next to each other and the different p_T^Z binning for the bin with maximum y^* leads to different slopes in the corresponding off-diagonal elements. To make the small migrations visible a logarithmic scale is chosen. The condition number for each is smaller than three.

normalized to the total number of events in the corresponding row in the reconstruction level dimension. This results in the migration matrix that encodes the full migration of the generator level observables to the reconstruction level observables.

The migration matrices obtained from filling the signal events generated for each of the individual data-taking periods are shown in fig. 5.35. The most significant entries are on the diagonal illustrating that there are only small migrations in phase space. Therefore, a logarithmic scale is chosen for visualization of the small off-diagonal contributions. Indeed, in this depiction it can be observed that migrations occur between neighboring bins in phase space. For the y_b-y^*-bin with highest y^* corresponding to the highest bins in the unravelled migration matrix, a different slope is observed in the off-diagonal elements. This is due to a different number of p_T^Z bins compared to the other y_b-y^*-bins leading

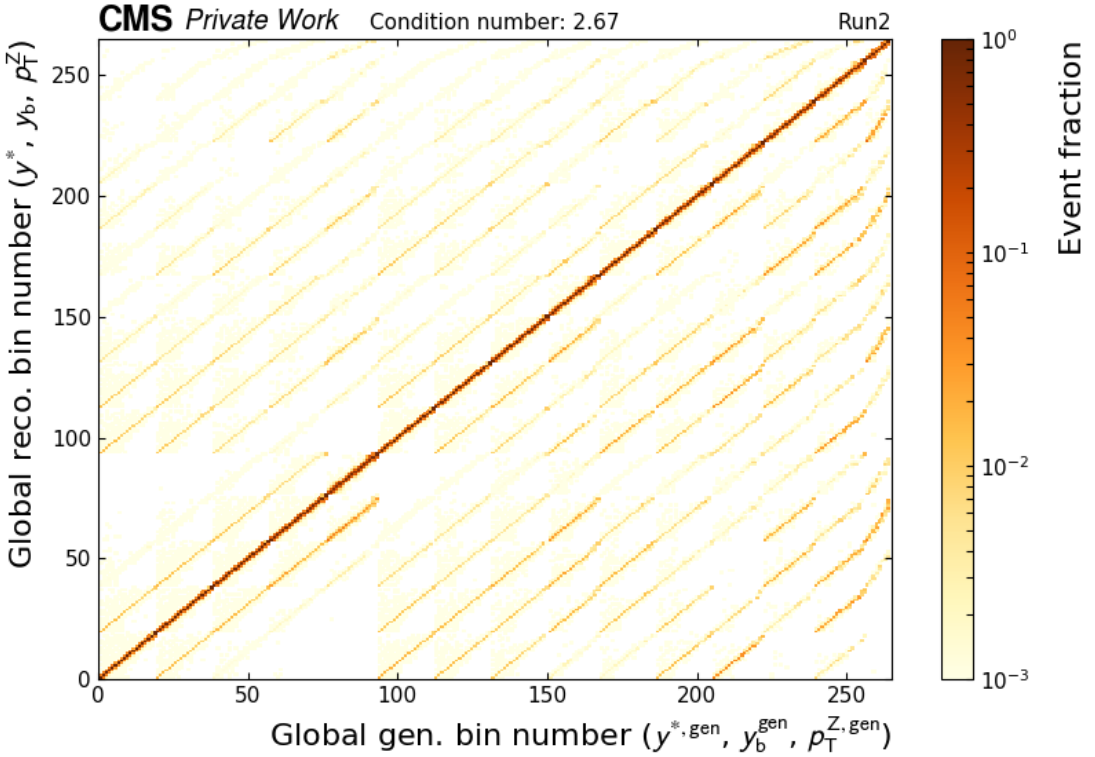


Figure 5.36: Visualization of the migration matrix for the unfolding of the combined Run 2 dataset. The most significant entries are on the diagonal of the matrix. Therefore, a logarithmic scale is used making the off-diagonal elements visible, which are dominated by neighboring bins in phase space. Note that due to the one dimensional representation of the phase space chosen in this visualization neighboring bins in y_b and y^* are not next to each other. At the edge of the matrix the p_T^Z binning in the corresponding y_b-y^*-bins is different leading to a different slope of the corresponding off-diagonal. The condition number is smaller than 3.

to a asymmetric submatrix with a different slope of the diagonal indicating regions of same p_T^Z. The general recommendation is to not include regularisation procedures for condition numbers smaller than ten. The condition number of the matrices is smaller than three, implying a well-conditioned inversion problem. All four matrices obtained for the respective data-taking periods are found to be similar, matching the observations in [89]. Therefore, a combined migration matrix from the individual simulations weighted by their respective luminosity for the corresponding data-taking period is constructed.

The migration matrix for the unfolding of the combined Run 2 data is shown in fig. 5.36. As expected, it shows the same features as the migration matrices constructed for the individual data-taking periods. It is mostly diagonal and well-conditioned.

Acceptance and Fakerate – There are events in the simulation, which are selected either on generation or reconstruction level. However, they are not selected in both leading. Consequently, the events are filled in respective under- and overflow bins for each y_b-y^*-p_T^Z-bin of the migration matrix. The under- and overflows in the normalized migration matrix in both dimensions i and j have a special role in the unfolding procedure. The contribution of events that solely pass the selections in reconstructed or generation level are assigned to these special bins in the two dimensional histogram.

Events that pass the selections at generation level but not on reconstruction level are considered a *loss*. Independent of the origin for this effect these events migrate outside the analysed phase space or do not pass the selection and quality criteria. Consequently, they contribute to the *acceptance* of the analysis. The acceptance is defined as the fraction of events that pass the selections on both generation and reconstruction level over the events that pass only the generation level selections for each y_b-y^*-p_T^Z-bin.

In reverse, events that pass the selections at reconstruction level but not on generation level are considered *fakes*. These events migrate into the analysed phase space or are selected by mistake due to suboptimal reconstruction. Consequently their contributions need to be treated as a background and are subtracted from the event yields input into the unfolding procedure. The corresponding *fakerate* is defined as one minus the fraction of events that pass the selections at both levels over the events that only pass the selections at reconstruction level.

As parts of the migration matrix the acceptances and fakerates are both constructed from the simulated signal events. To estimate the stability over time the acceptances and fakerates are depicted differentially in p_T^Z-bins but inclusive in y_b-y^* in fig. 5.37 following [89]. No significant deviation between the individual data-taking periods is found.

The acceptances and fakerates differentially in p_T^Z-bins but inclusive in y_b-y^* for the combined Run 2 dataset are shown in fig. 5.38. A detailed split into the individual y_b-y^*-p_T^Z-bins is shown in fig. A.20.

For low p_T^Z, close to the selection boundary the fakerates are at their maximum. They converge for high p_T^Z towards zero and the rate of convergence drops for higher rapidity bins. This observation is expected since the reconstruction and identification of muons are worst for soft and high-rapidity regions of phase space and improve with higher p_T and lower rapidities. The same behaviour is expected for the acceptances, since for the same reasons the reconstruction efficiencies follow the same trend. Consequently, the lowest acceptances are observed for small p_T^Z. They increase for growing p_T^Z until reaching a plateau at approximately 80%. For high p_T^Z the acceptances drop again. This is due to migrations of events outside of the analysis phase space. The same drop in acceptance is observed for the outer y_b-y^*-bins.

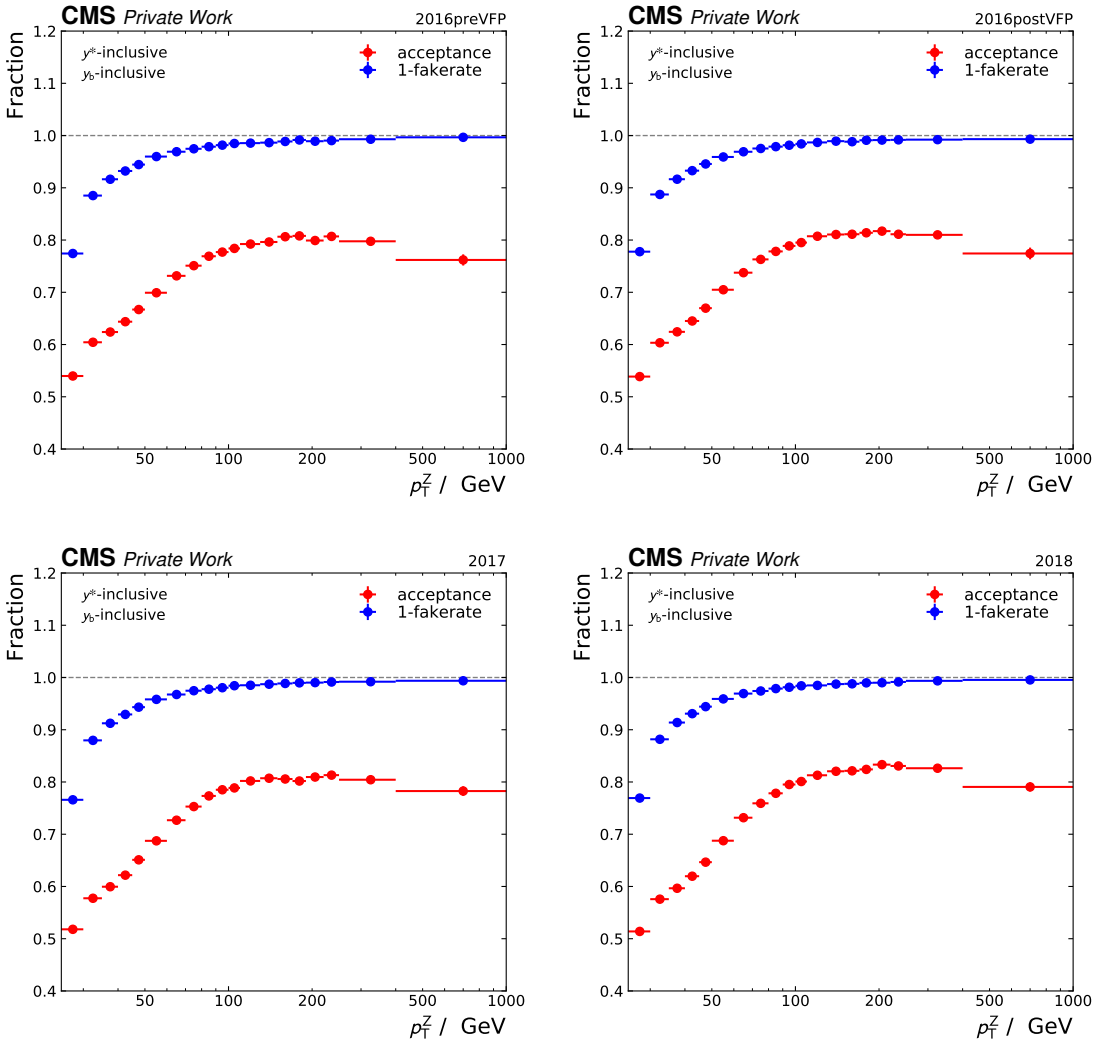


Figure 5.37: Acceptance and 1-fakerate (see section 5.5.2) inclusive in y_b-y^* constructed for the individual data-taking periods 2016preVFP, 2016postVFP, 2017, and 2018, (see sections 5.2 and 5.3) respectively.

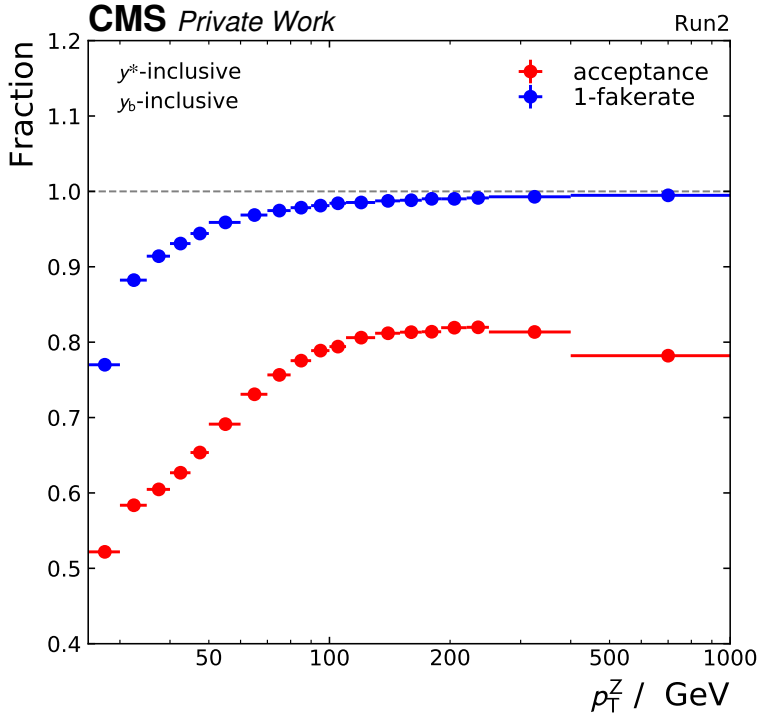


Figure 5.38: Acceptance and 1-fakerate inclusive in y_b-y^* constructed for the combined Run 2 data. The full split for all y_b-y^*-p_T^Z-bins is shown in fig. A.20. The fakerate is maximal at low p_T^Z and converges towards 0 for high p_T^Z. The acceptance is minimal at low p_T^Z and reaches a plateau for high p_T^Z. Towards the boundaries of the analysed phase space the acceptances drop again and the convergence of the fakerates is slowed.

5.5.3 Cross Checks

Since unfolding presents an ill-posed problem and relies on the correct description of reality by the utilised theoretical models to construct the migration matrix systematic biases can be introduced. Therefore, the consistency and stability of the unfolding procedure is checked before the final application on data and the comparison of the unfolded results to theoretical predictions.

5.5.3.1 Unfolding Closure

A consistency check is performed by unfolding the event yields at reconstruction level of the simulation used to fill the migration matrix and comparing it to the predicted cross sections at generation level. When the unfolding procedure works as intended a perfect agreement between the two sets of observables is expected.

Figure 5.39 shows the comparison of the unfolded simulated event yields with the corresponding predictions for the central y_b-y^*-bin and the two bins with maximal y_b and y^*. The closure for all y_b-y^*-bins is shown in fig. A.21. A perfect agreement between the two distributions can be observed for the whole analysed phase space. Consequently, it can be confirmed that the unfolding procedure behaves as intended and is consistent.

5.5.3.2 Unfolding Bias Estimation

The unfolding procedure relies on the proper simulation of detector and reconstruction effects in order to produce a migration matrix, which correctly describes the transformation of the true observables into the reconstructed ones. When a bias exists in the simulation, the application of the reverse transformation on the measured yields in data becomes incongruous.

To estimate such a systematic bias introduced by the choice of the generator in the creation of the simulated events the unfolding procedure is repeated twice. In each unfolding the same yields obtained from data are unfolded using a migration matrix obtained from simulations obtained from different generators. On generation level the predicted cross sections are significantly different assuring that if a bias is present the effect on the unfolded cross sections will be significant.

A comparison of the two unfolded results is shown in fig. 5.40 for the central y_b-y^*-bin and the two bins maximal in y_b and y^*. The comparisons for all y_b-y^*-p_T^Z-bins are illustrated in fig. A.22. No significant deviations between the two sets of observables within statistical uncertainties (see section 5.6.1) are observed. This confirms that there is no systematic bias due to the choice of the generator present and the unfolding procedure is stable. Consequently, there is no need for assigning an additional uncertainty related to the choice of the simulated sample.

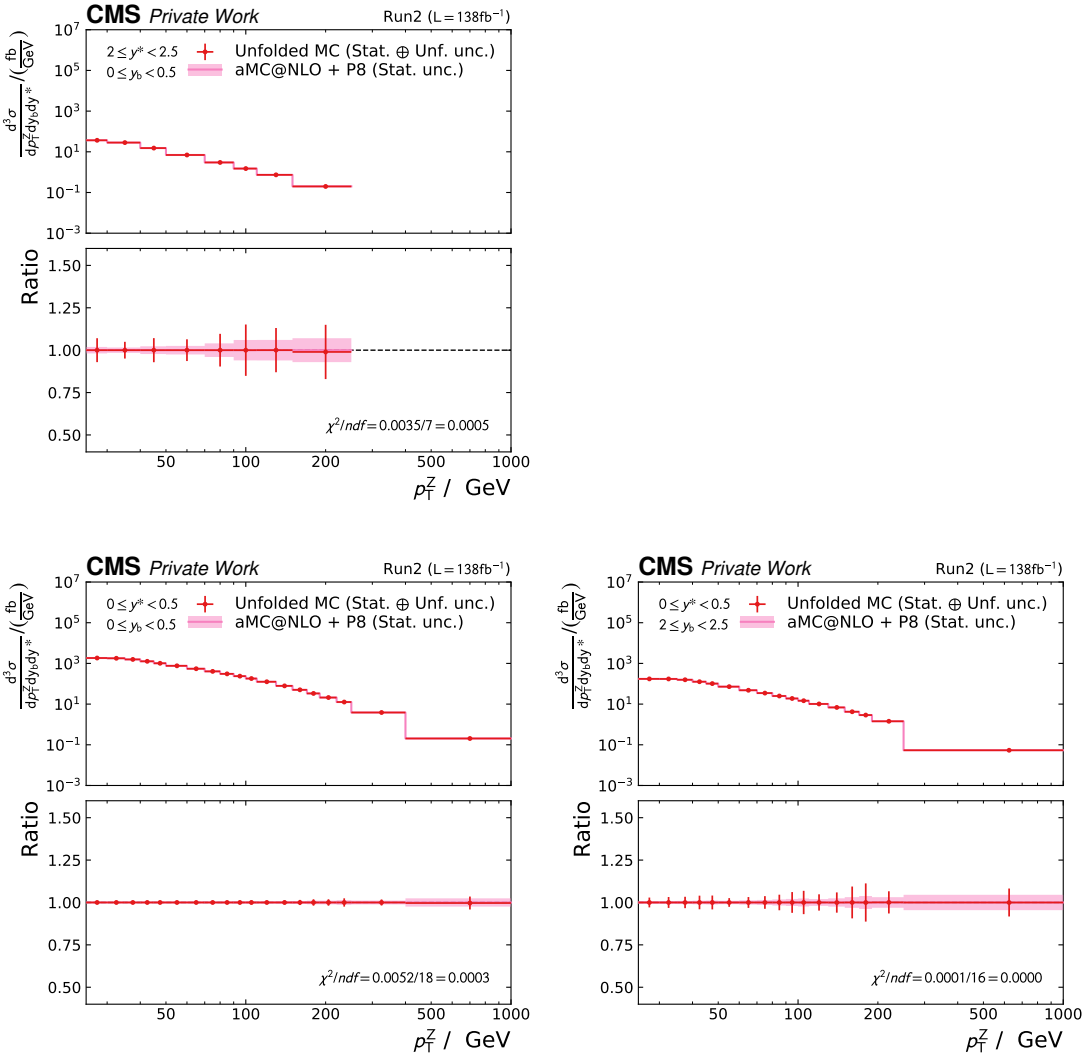


Figure 5.39: Closure of the unfolding procedure used for the combined Run 2 data for the central and two extreme y_b-y^*-bins. The migration matrix constructed from the combined set of simulated events is used for performing the unfolding on the simulated signal yields on reconstruction level. The unfolded results (red points) with statistical uncertainties are compared to the corresponding predictions at generation level with statistical uncertainties (pink band). The two sets are in perfect agreement.

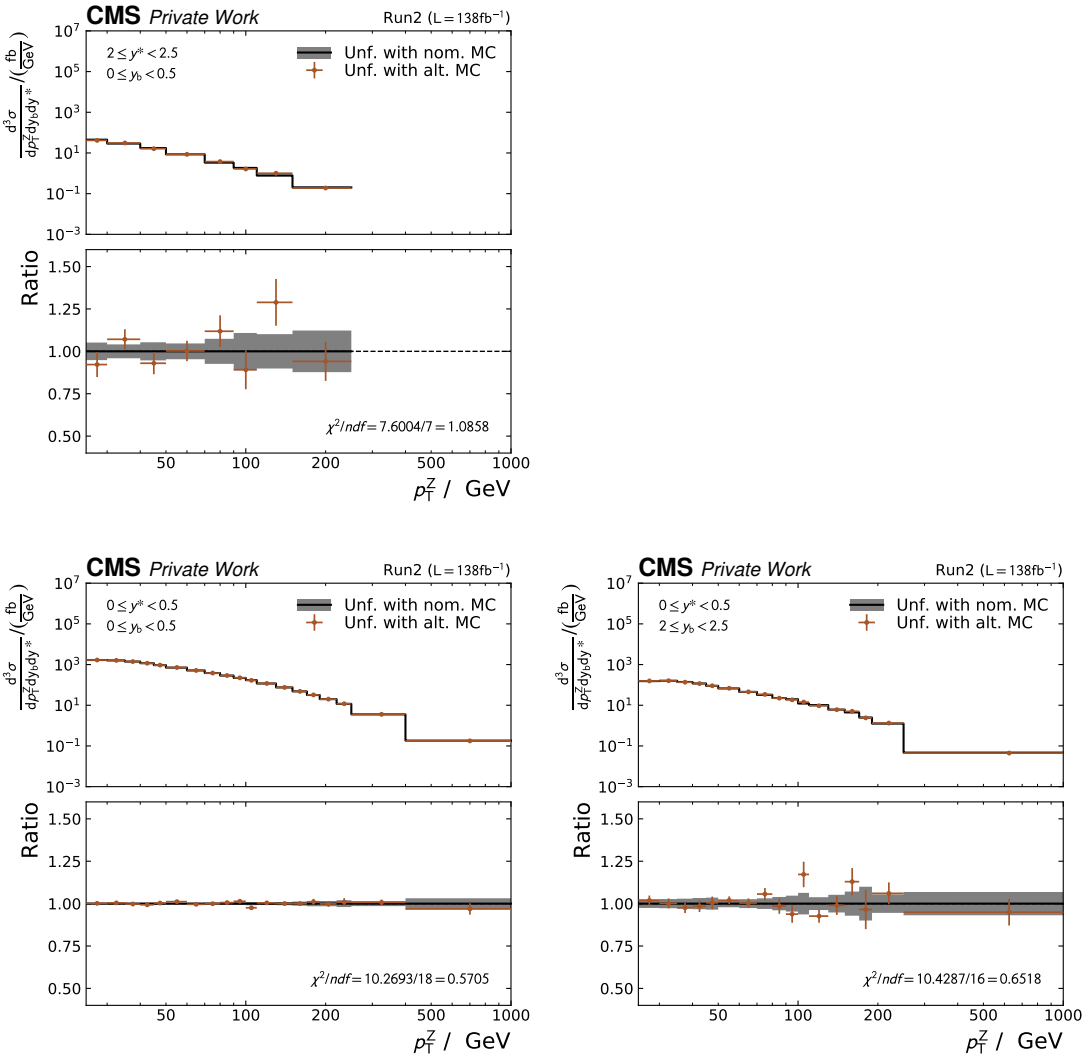


Figure 5.40: Check of the systematic bias introduced by choice of a specific simulation in the unfolding procedure used for the combined Run 2 data for the central and two extreme y_b-y^*-bins. Two migration matrices are constructed from the combined set of simulated events from two distinct generators. The two sets of unfolded cross sections obtained from using the two alternative migration matrices are compared. The results obtained with statistical uncertainties from the nominal simulation used in this analysis are shown as a gray band. The results from the alternative are shown as orange points with whiskers showing the corresponding statistical uncertainties. No significant deviations between the two sets are observed apart from statistical fluctuations.

5.6 Uncertainties

The utilised reconstruction and analysis methods in this thesis are subject to statistical and systematic effects. These are propagated to the measured unfolded results presented in this thesis. Consequently, uncertainties originating in these effects need to be propagated as well. In this section, the estimation methods of the various uncertainties on the unfolded results and the combination of the uncertainties assigned to the individual sources to create a total uncertainty are described.

5.6.1 Statistical Uncertainties

Two different types of statistical uncertainties enter the unfolding procedure.

The first set of statistical uncertainties are the ones assigned to the measured event yields in data for each y_b-y^*-p_T^Z-bin s which follow simple Poisson statistics. Before unfolding the uncertainties for each bin are uncorrelated between bins. Consequently, the corresponding covariance matrix $\mathbf{V_{ss}}$ is diagonal. After applying the unfolding procedure the resulting covariance matrix for the unfolded results $\mathbf{V_{tt}}$ is not diagonal anymore. Due to the mitigation of migrations and acceptance/efficiency effects the unfolded cross sections t are statistically correlated. The resulting correlated statistical uncertainties are labelled as *statistical uncertainty* in the following.

Another set of statistical uncertainties originate in the limited number of simulated events used for the construction of the migration matrix $\mathbf{R}$. Following the statistics of weighted Poisson events [91] statistical uncertainties are constructed for each entry in the migration matrix from the sum of the squared weights w_i of events i filled in the corresponding bin:

$$\sigma^2_{\text{stat, bin}} = \sum_{i \in \text{bin}} w_i^2 \tag{5.24}$$

Consequently, each bin of the migration matrix is assigned a statistical uncertainty. These uncertainties are propagated in TUnfold (see section 5.5.1) to the unfolded results leading to an additional contribution to $\mathbf{V_{tt}}$. This type of statistical uncertainty is labelled as *unfolding uncertainty* in the following.

Both statistical and unfolding uncertainties are shown for the central and extreme y_b-y^*-bins in fig. 5.41 and for all y_b-y^*-p_T^Z-bins in fig. A.23. The number of events in both, data and simulation, are the smallest for high p_T^Z, y^*, and y_b and maximum in the low p_T^Z and central regions of phase space. Consequently, the uncertainties are the largest for high p_T^Z and forward regions with high rapidities. These regions of the analysed phase space are dominated by the statistical and unfolding uncertainties.

Statistical uncertainties on event yields constructed from the simulated events and not propagated through the unfolding procedure are estimated with eq. (5.24). If necessary, additional normalization scaling factors are multiplied to construct observables

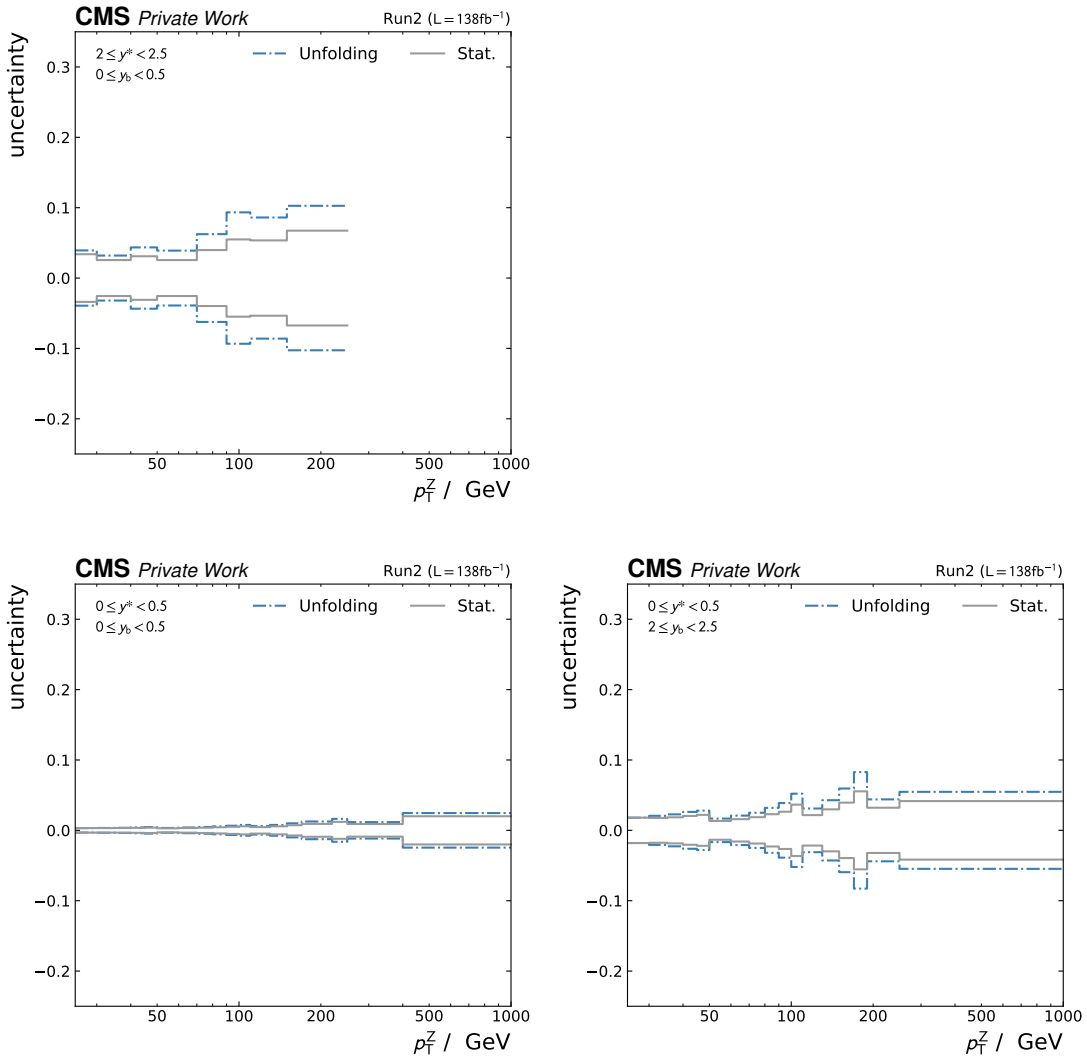


Figure 5.41: Uncertainties originating from the limited number of events in data (statistical uncertainty) (gray) and the limited number of events in simulation utilised for the construction of the migration matrix (unfolding) (blue) for the unfolded cross sections obtained for the combined Run 2 data for the central and two extreme y_b-y^*-bins. Both uncertainties are derived by the TUnfold package. Since the number of events are the smallest for high p_T^Z, y^*, and y_b in both, data and simulation, the uncertainties are largest for high p_T^Z and rapidities. They dominate in these regions of the analysed phase space.

like cross sections from the event yields. These uncertainties are labelled with *statistical uncertainty* as well.

5.6.2 Systematic Effects and Uncertainties

The reconstruction of the measured objects and their assigned kinematic variables are subject to uncertainties (see section 4.2). The same holds for the applied corrections and quality criteria (see section 5.1.2) and the analysis methods. The uncertainties originate in the limited knowledge of the exact interactions of the collision products with the detector, the complex nature of the subsequent reconstruction, the calibration of the reconstruction methods, and the choice of analysis methods and estimates. In all these steps assumptions on the utilised models and their parameters are made, which possibly introduce a systematic bias. These biases are typically estimated by model or parameter variations. Additionally, in the calibration of these models the validity is typically assessed in dedicated studies, which are subject to statistical uncertainties but ultimately assigned as part of the systematic uncertainty of the corresponding source.

The effects of the systematic biases and the statistical uncertainties on the analysed observables are both propagated to the analysed observables as *systematic uncertainties*. One set of systematic uncertainties is derived in this analysis for each systematic source that comprises the estimation of the background contributions, the luminosity, the efficiency scale factors in the muon reconstruction, selection, and identification, the correction of the L1 prefiring effect, the identification of PU jets, and the calibration of the jet energy scale and resolution.

The analysed dataset is composed of the data measured in four individual data-taking periods with dedicated generation and simulation data.As a consequence, correlations between the uncertainties on the contributions of the individual datasets have to be accounted for.

5.6.2.1 Background Estimation

The contribution of background to the measured event yields in data is estimated using the simulated events for the identified background processes (see section 5.3.1). The estimated background contributions to the event yields in the analysed phase space bins are subtracted from the corresponding yields measured in data prior to unfolding.

To estimate the effect of an imperfect estimation of the backgrounds and its effects on the unfolded cross sections the normalization of the backgrounds is varied by $\pm 50\%$ and the unfolding procedure is repeated for each of the varied input data yields. Since the background predictions for each individual data-taking period are generated using the same underlying generators full correlation is assumed. Therefore only one respective up and down variation of the background predictions for the full Run 2 dataset is performed and propagated through the unfolding procedure. There are full correlations

between individual bins expected. The resulting two unfolded results per y_b-y^*-p_T^Z-bin are interpret as the corresponding background uncertainty. The background uncertainty for the central y_b-y^*-bin and the two bins with maximal y_b and y^* respectively is shown in fig. 5.42 and for all bins in fig. A.24.

5.6.2.2 Luminosity Uncertainty

The integrated luminosity is measured by the CMS collaboration utilising various luminometer calibrated with Van-der-Meer scans (see section 3.1). The resulting measurements of the luminosities for the data-taking years 2016, 2017 and 2018 are 19.52 fb^{-1}±1.2%, 16.81 fb^{-1}±1.2% [92], 41.48 fb^{-1}±2.3% [93], and 59.83 fb^{-1}±2.5% [94]. However, due to the way the luminosity is measured correlations between the individual uncertainties for each data-taking period exist and need to be taken into account. The correlation matrix derived from inputs given by the CMS collaboration [135] reads

$$\begin{pmatrix} 1 & 0 & 0.2 & 0.41 \\ 0 & 1 & 0.2 & 0.41 \\ 0.2 & 0.2 & 1 & 0.34 \\ 0.41 & 0.41 & 0.34 & 1 \end{pmatrix} \tag{5.25}$$

with the dimensions ordered in the sequence 2016preVFP, 2016postVFP, 2017 and 2018. This results in a total luminosity uncertainty for the combined Run 2 data set of 1.6%.

The uncertainty on the luminosity is propagated to the unfolded cross sections. First, the nominal migration matrix for each individual data-taking period is varied by the correlated and uncorrelated proportions of the respective luminosity uncertainty in each bin and the unfolding procedure is repeated. Afterwards, the individually varied unfolded results are combined to obtain the luminosity uncertainty on the full analysed dataset. The luminosity uncertainty for the central and two extreme y_b-y^*-bins and each bin is shown in fig. 5.42 and fig. A.24, respectively.

5.6.2.3 Muon Efficiencies

To estimate the effect of the uncertainty on the muon efficiency scale factors provided by the CMS collaboration (see section 5.1.2.3), the scale factors are varied to the upper and lower bounds of their uncertainty and the creation of the migration matrix is repeated for each variation. The uncertainty contributions of each data-taking period are assumed to be fully correlated allowing to simply add their contributions following the Poisson statistics of weighted events (see section 5.6.1). The individual y_b-y^*-p_T^Z bins are assumed to be fully correlated. The unfolding of the measured data yields is repeated with each alternative response matrix. The corresponding difference between the nominal and alternative unfolded results are adopted as the uncertainty associated with the muon reconstruction, labelled *muon scale factor (SF) uncertainty* in the following.

The resulting muon SF uncertainties are shown for the central y_b-y^* and the two bins

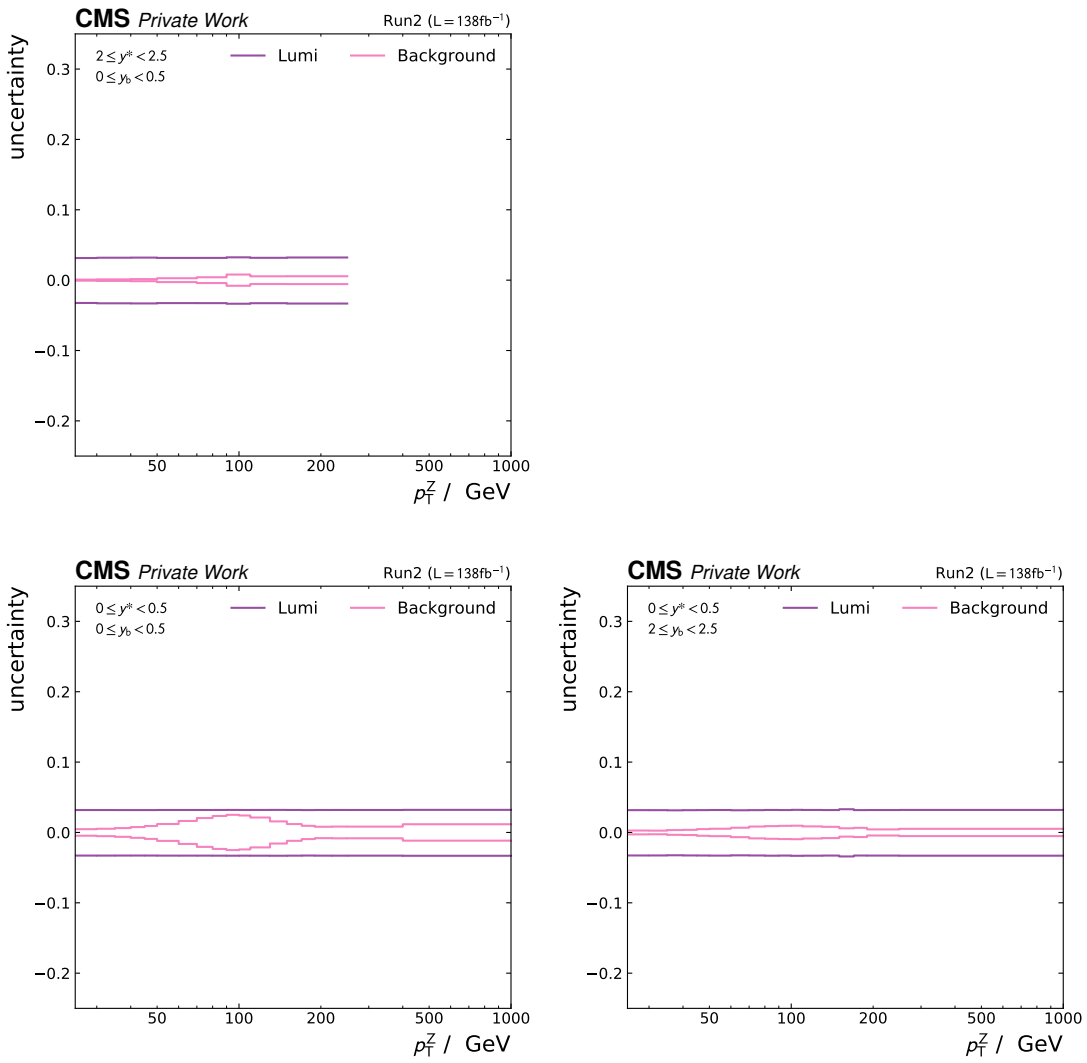


Figure 5.42: Background (pink) and luminosity (violet) uncertainties for the unfolded cross sections obtained for the combined Run 2 data for the central and two extreme y_b-y^*-bins. The luminosity uncertainty is derived by varying the nominal migration matrix by all correlated and uncorrelated proportions for the individual data-taking periods and combining the varied unfolded results. It is the same over the whole phase space. The background uncertainty is derived from varying the background contributions subtracted from the measured data yields and propagating each variation through the unfolding. It is largest for p_T^Z close to the mass of the Z boson, where the background contribution is the largest but always smaller than the luminosity uncertainty.

with maximal y_b and y^* respectively in fig. 5.43 and for the whole analysed phase space in fig. A.25. The muon SF uncertainty has only a marginal dependence on p_T^Z and increases slightly towards higher p_T^Z, y^* and y_b. It is considerably smaller than the luminosity uncertainty in all bins.

5.6.2.4 L1 Prefiring Correction

The uncertainty associated to the correction of the L1 prefiring is estimated by varying the corresponding event weights (see section 5.1.2.3) according to the advertised upper and lower uncertainty values in the provided probabilities. The uncertainty contributions of each data-taking period are assumed to be fully correlated allowing to simply construct a single migration matrix for the combined dataset for each variation. The individual y_b-y^*-p_T^Z bins are assumed to be fully correlated. These two varied sets of simulated events are then used to construct alternative migration matrices, respectively, which are subsequently used to unfold the measured event yields in data. The difference between the nominal and the corresponding alternative unfolded cross sections per bin are defined as the upper and lower bounds of the *L1 prefiring uncertainty*.

The L1 prefiring uncertainties are shown for all y_b-y^*-p_T^Z bins in fig. A.25 and selected central and extreme y_b-y^*-bins in fig. 5.43. The uncertainty increases with higher p_T^Z, y^* and y_b but is smaller than the luminosity uncertainty even in the most extreme bins.

5.6.2.5 PU Jet Identification

The *PU jet ID (also labelled as PUJetID) uncertainty* is as well estimated by varying the obtained event weights for the PUJetID efficiency corrections in the simulationset (see section 5.1.2.3) according to the upper and lower uncertainty shifts in the scale factors. The uncertainty contributions of each data-taking period are assumed to be fully correlated allowing to simply add their contributions following the Poisson statistics of weighted events. The individual y_b-y^*-p_T^Z bins are assumed to be fully correlated. Subsequently, the migration matrices for the full analysed dataset are constructed and the unfolding of the measured data yields is performed for each of the two variations. The difference between the resulting alternative and nominal unfolded cross sections are interpreted as the upper and lower bounds of the PUJetID uncertainty.

The obtained uncertainties are depicted for the central and extreme bins in y_b-y^* in fig. 5.43 and for all y_b-y^*-p_T^Z bins in fig. A.25. The PUJetID uncertainty is largest for low p_T^Z and decreases for higher p_T^Z. This is expected since the PUJetID is only applied on jets with transverse momenta $p_T^{\text{jet}} < 50\,\text{GeV}$ and the p_T^Z is correlated with p_T^{jet}. For high rapidities y_b and y^* it is larger which can be explained due to higher contributions of PU in the forward directions of the detector. However, it is for most bins, except the lowest in p_T^Z and highest in y_b and y^*, smaller than the luminosity uncertainty.

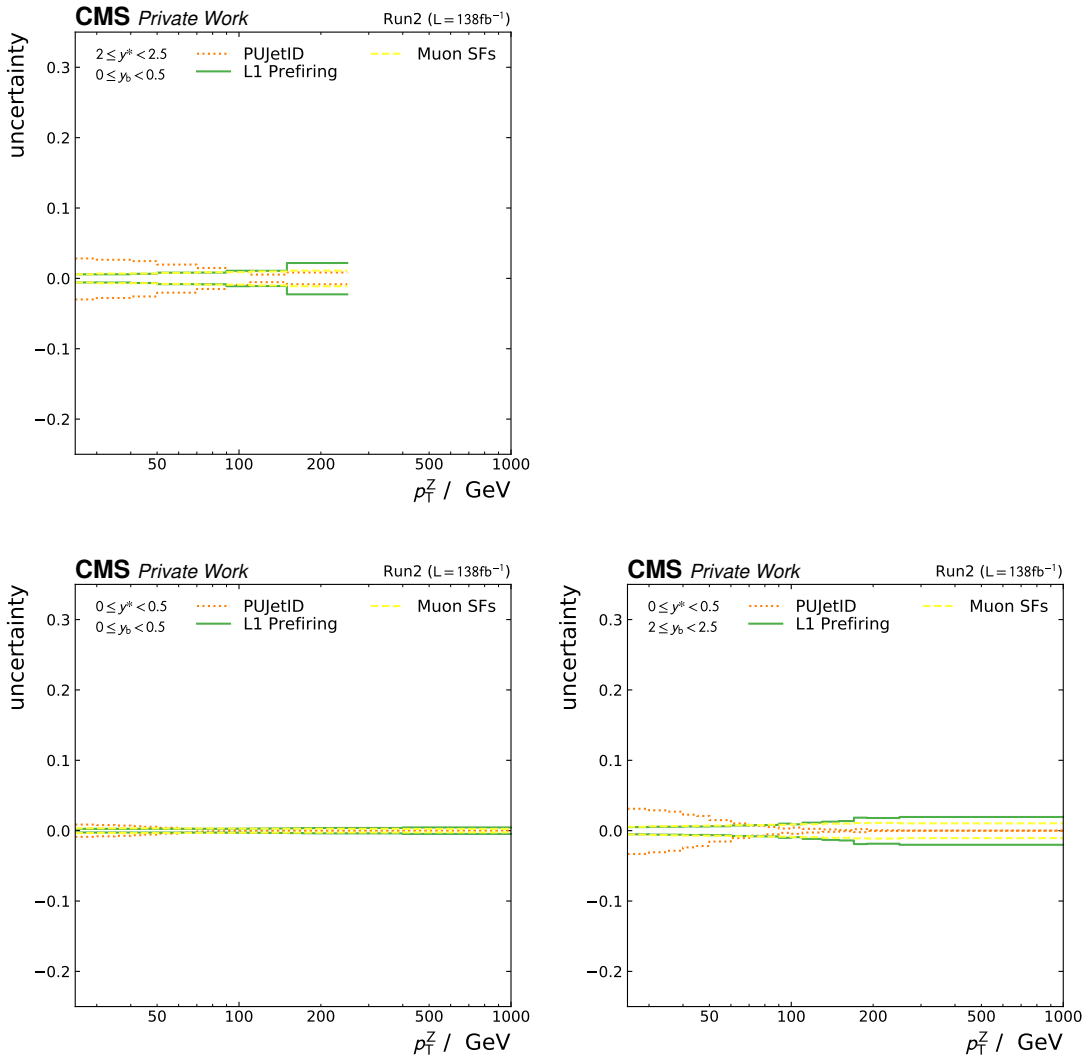


Figure 5.43: Muon scale factor (yellow), L1 prefiring (green), and PU jet identification (orange) uncertainties for the unfolded cross sections obtained for the combined Run 2 data shown for the central and two extreme y_b-y^*-bins. They are derived by constructing alternative migration matrices from the events reconstructed with variations of the muon scale factors, the L1 prefiring correction weights, and the PUJetID efficiency correction weights, respectively within their corresponding uncertainties. For each the unfolding of the measured data yields is repeated and the difference between the nominal and the alternative unfolded cross sections is interpret as the corresponding uncertainty. The PUJetID uncertainty contributes mostly in the low p_T^Z region and decreases towards high p_T^Z. The L1 prefiring uncertainty increases with p_T^Z. The muon scale factor uncertainty has only a slight dependence on p_T^Z. They are significantly smaller than the luminosity uncertainty in all analysed bins except for the PUJetID uncertainty which reaches similar orders of magnitude for the smallest p_T^Z and high rapidities.

5.6.2.6 Jet Energy Resolution Correction

The *jet energy resolution (JER) uncertainty* on the measured unfolded cross sections is derived by repeating the energy resolution correction of the jets in simulation by varied scale factors obtained from their upper and lower uncertainty boundaries. The uncertainties assigned to individual y_b-y^*-p_T^Z bins are assumed to be uncorrelated with other bins. The uncertainty contributions of each data-taking period are assumed to be fully correlated. Therefore, the influences of the JER variations applied on a single of the four data-taking periods are separately derived. The variations are only applied on the events assigned to the corresponding data-taking period, while all other events are unchanged. Consequently, two times four new alternative migration matrices are constructed and the unfolding of the measured event yields in data is repeated. The boundaries of the JER uncertainty for each data-taking period are defined as the differences of the two corresponding alternative unfolded results with the nominal unfolded result. Subsequently, the total JER uncertainty is constructed as the quadratic sum of the four individual contributions.

The total JER uncertainties are shown for all analysed bins in fig. A.26 and for central and the two extreme bins in y_b-y^* in fig. 5.44. The JER uncertainty contributes the most in the low p_T^Z region and decreases towards high p_T^Z. This is expected, since the jet energy resolution is best for high p_T^{jet} [79] and p_T^Z is correlated with p_T^{jet}. Also at higher p_T^Z and p_T^{jet} the event selection is less influenced by variations of the jet energy. It also increases with higher rapidities matching the expectation due to the bigger uncertainties on the JER corrections in forward direction [79]. However, the total size of the JER uncertainties is compatible with the Muon SF and L1 prefiring uncertainties.

5.6.2.7 Jet Energy Scale Correction

Similar to the JER uncertainties the *jet energy scale (JEC) uncertainties* on the measured unfolded cross sections are estimated. The correction factors applied on the jet energy are subject to systematic uncertainties (see section 5.1.2.3) that are provided by the CMS collaboration. A full set of 26 individual sources and corresponding uncertainties contributing to the uncertainty assigned to the correction factors with different correlations between the uncertainties in the individual data-taking periods is provided. However, in this work, only the combined uncertainty on the correction factors is considered. A full breakdown of the effect of each individual JEC uncertainty source is left for future studies. The total uncertainty on the correction factors is assumed to be fully correlated between data-taking periods and between y_b-y^*-p_T^Z-bins. This is a conservative estimate since the individual sources are a mix of fully, partially and uncorrelated JEC uncertainties. By combination of the individual sources with all correlations taken into account a decrease in the resulting total JEC uncertainty is expected.

Consequently, the JEC uncertainties are estimated by varying the jet energy correction factors according to the upper and lower uncertainty and applying them on the simulated

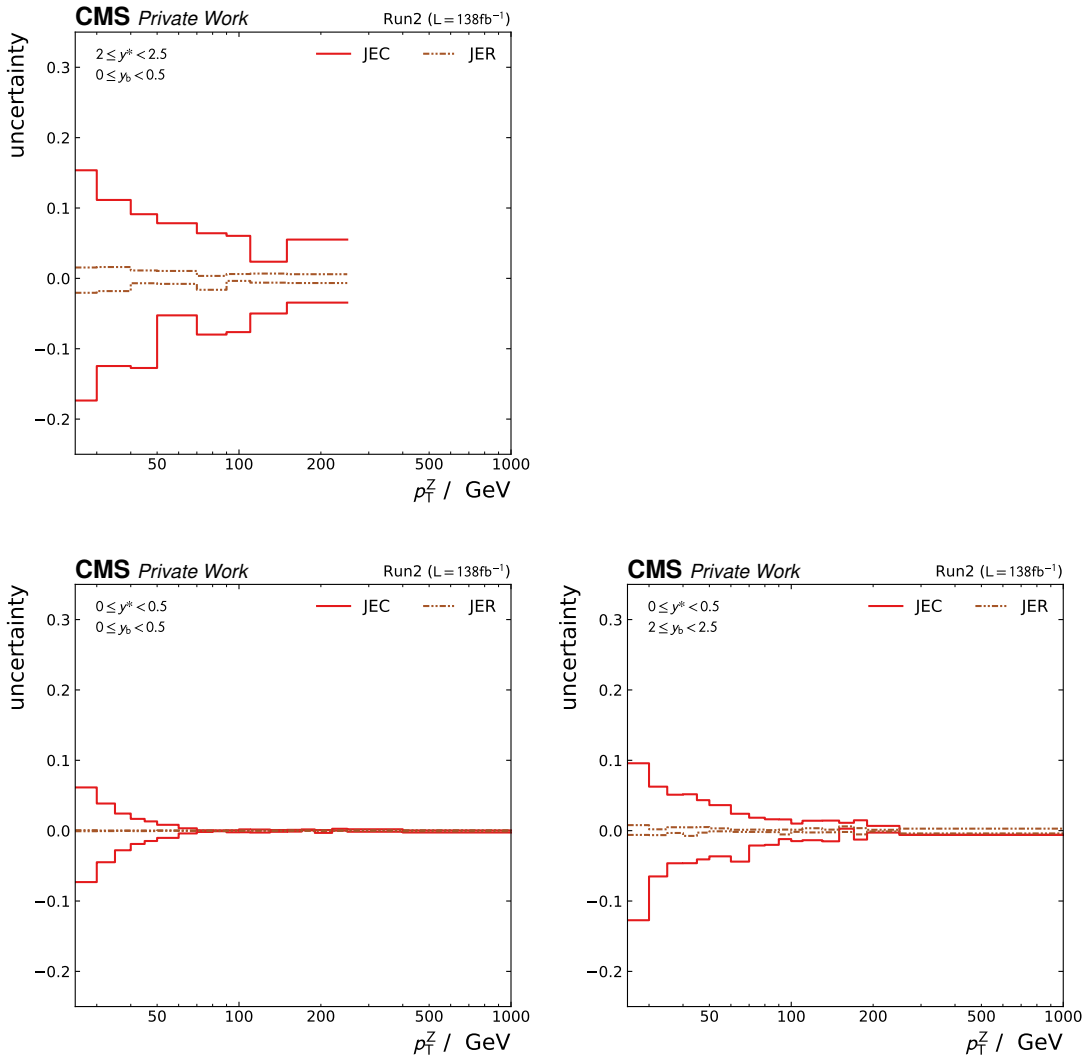


Figure 5.44: Jet energy resolution (JER) (brown) and jet energy scale (JEC) (red) uncertainties for the unfolded cross sections obtained for the combined Run 2 data for the central and two extreme bins in y_b-y^*. They are derived by constructing alternative migration matrices from the events with jet energies corrected with scale factors varied respectively within their corresponding uncertainties. Since the JER is assumed to be fully uncorrelated between data-taking periods the variations of each of the four periods are exclusively leading to total four times two variations. For each the unfolding of the measured data yields is repeated and the difference between the nominal and the alternative unfolded cross sections is interpret as the corresponding uncertainty. The total JER uncertainty is constructed as the quadratic sum of the four contributions in each bin. The JER uncertainty contributes the most in the low p_T^Z region and decreases towards high p_T^Z. The JEC uncertainty shows the same behaviour but is an order of magnitude larger. While the JER uncertainty is significantly smaller than the luminosity uncertainty in all analysed bins the JEC uncertainty dominates for small p_T^Z.

jets. From these two variations an alternative migration matrix is constructed, which are subsequently applied on the measured data yields to construct two alternative results. The upper and lower bounds on the JEC uncertainties on the unfolded cross sections is obtained from the differences between the two alternative results and the nominal ones.

They are shown for the central and two extreme bins in y_b-y^* in fig. 5.44 and for all y_b-y^*-p_T^Z-bins in fig. A.26. Since the uncertainties on the JEC correction factors are largest for low p_T^{jet} [79] and p_T^{jet} and p_T^Z are correlated in this analysis, the estimated JEC uncertainties are largest for small p_T^Z and decrease with increasing p_T^Z. Additionally, the selection of events is less prone to variations of p_T^{jet} when p_T^{jet} is much bigger than the according selection (see section 5.1.2.4) leading to a smaller influence at high p_T^Z. For increasing y_b and y^* the uncertainty increases as well. This is expected since in the forward region with high η the uncertainties on the correction factors increases as well [79]. The JEC uncertainties dominate the total uncertainties in the low p_T^Z regions of the analysed phase space.

5.6.3 Total Uncertainty

The uncertainties estimated above for the individual sources are assumed to be independent from each other and therefore uncorrelated. This is not particularly true since the estimation procedures rely in parts on the analysis of the same data and simulation. As a consequence, correlations between the individual sets of uncertainties exist and neglecting these correlations and assuming the sources are fully uncorrelated leads to an conservative estimation of the total uncertainty. Assuming separately normal distributed distributions of the estimated uncertainty bounds, the total uncertainty is estimated as the quadratic sum of the bounds of the individual sources in each bin.

The total uncertainties together with all its comprised contributions are shown for the central and two extreme bin in y_b-y^* in fig. 5.45. The uncertainties for all cross sections are depicted in fig. A.27. In these comparison plots it can be directly observed that the JEC uncertainties dominate for the low p_T^Z regions of phase space and the unfolding and statistical uncertainties dominate for high p_T^Z. For the central rapidity bins the total uncertainties are smaller than 5% for most bins. In high y_b regions the total uncertainty is still below 10% for most bins. For high y^* it reaches up to approximately 17% in the bin with the worst precision for small p_T^Z. A thorough uncertainty break down of the JEC uncertainty sources with a statistical combination taking all correlations into account is expected to reduce the uncertainty for low p_T^Z.

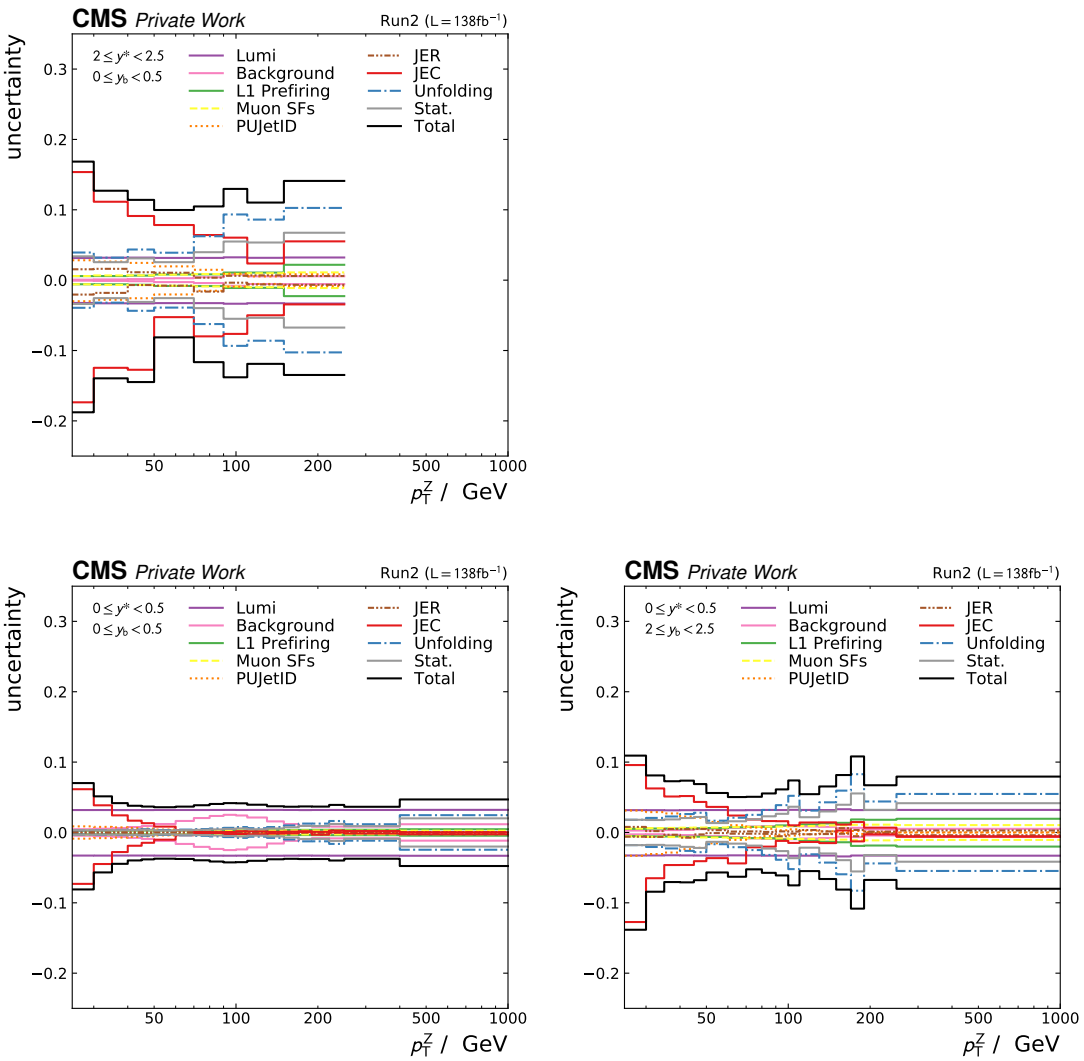


Figure 5.45: Overview of all considered uncertainties including uncertainties originating in the limited statistics in data (statistical uncertainty) (gray), originating in the limited statistics in simulation utilised for the construction of the migration matrix (unfolding) (blue), muon scale factor (yellow), L1 prefiring (green), PU jet identification (orange), jet energy resolution (JER) (brown), and jet energy scale (JEC) (red) uncertainties for the unfolded cross sections obtained for the combined Run 2 data for the central and two extreme bins in y_b-y^*. The total uncertainty (black) is defined as the quadratic sum of each individual source's contribution. The low p_T^Z region of phase space is dominated by the JEC uncertainties. In the high p_T^Z region the statistical and unfolding uncertainty dominate.

5.7 Comparison of Measured Cross Sections to Theoretical Predictions

Finally, the measured unfolded cross sections are compared to theoretical predictions. The theoretical predictions are derived from the generator level events of two datasets created with the MadGraph5_aMC@NLO [97, 98] and Pythia 8 [29] event generators at LO and NLO accuracy in perturbative QCD.

The first sample generates the production of dimuon events inclusive in the number of jets in proton-proton collisions at a center-of-mass energy of 13 TeV. The hard interaction is generated by MadGraph5_aMC@NLO matched with the Pythia 8 parton shower using the MC@NLO matching method [30] producing events with a pair of oppositely charged muons and up to four partons at LO accuracy utilising the PDF set NNPDF 3.1 [43]. The contributions by the distinct parton multiplicities are merged using the MLM jet merging method [136]. After the parton shower, the hadronization and UE are generated by Pythia 8 as well. As such, it approximates the fixed-order calculations for the fully differential production cross section of dimuon events plus one jet at N^3LO accuracy containing all real but missing the virtual corrections with LL resummation added by the parton shower. Non-perturbative corrections are included by the generation of hadronization and UE. The predictions created with this sample are labelled *MLM + P8* or *LO*, matching the perturbative order of the included matrix elements of the hard process at all multiplicities, in the following.

The second sample is the sample used for the generation of the signal events (see section 5.3.1). In the hard interaction the production of one pair of oppositely charged muons plus up to two jets at NLO accuracy utilising the PDF set NNPDF 3.1 [43] are generated. Consequently, this sample approximates the fixed-order calculations for the fully differential production cross section of dimuon events plus one jet at NNLO accuracy, as well, containing all real but only one-loop virtual corrections. It misses the two-loop virtual corrections to reach full NNLO accuracy. The parton shower adds a LL resummation. Non-perturbative corrections are included by the generation of hadronization and UE. Predictions generated by this sample are labelled *aMC@NLO + P8* or *NLO* in the following.

Both samples are normalized to match the inclusive cross section prediction obtained with FEWZ [100–103] of $6077.22\,\mathrm{pb}^{-1} \pm 2\%$ for the Drell-Yan process at NNLO accuracy in QCD and NLO accuracy in EW for the fiducial phase space of the samples and utilising the same PDF set NNPDF 3.1 [43] as used in both. The uncertainty on this prediction of the inclusive cross section includes an estimation of the impact by uncertainties on the utilised PDF, uncertaintiies on the fragmentation and regularisation scales (see section 2.2.5), and statistical uncertainties related to the MC integration (see section 2.2.1) of the phase space. The uncertainties in the two shown samples include the statistical uncertainties related to the limited number of generated events (see section 5.6.1), the

uncertainty on the inclusive cross section for the process, and parton shower uncertainties (see section 2.2.5) and the uncertainty on the inclusive cross section prediction.

The measured cross sections compared to the theoretical predictions are shown for the central and two extreme bins in y_b-y^* in fig. 5.46 and for all y_b-y^*-p_T^Z-bins in fig. A.28. The LO cross section predictions are not able to capture the dependency of the measured cross sections on p_T^Z. Additionally, the normalization of the LO predictions is too low in the analysed phase space. The NLO predictions match the measured cross sections better. The shape matches in most y_b-y^*-p_T^Z-bins within the uncertainties. However, an offset in the normalization just at the edge of matching within the uncertainties is observed in most y_b-y^*-bins.

This difference in normalization of predicted cross sections compared to the measured cross sections shows a y^*-dependence. While the corresponding normalization shift for both samples is maximal for small y^* it decreases with growing y^*. No systematic dependence on y_b is observed. These observations match the ones made in the comparison of the measured and predicted event yields in section 5.4 depicted in fig. 5.34.

Finding the origin of the y^* dependence renders further studies necessary. The observed trend, however, indicates a systematic bias in the models or respective tunes used for the theoretical predictions, or missing higher order perturbative corrections. For instance, predictions at full NNLO accuracy in perturbative QCD that include the missing virtual corrections are expected to change the relative contributions of individual initial state partons to the partonic cross section (see section 2.2.2.1). The initial states consist of partons drawn from flavour-dependent PDFs. By a change in the initial state a change of the assigned momentum fractions and therefore the final state particles' rapidity distributions is expected. Consequently, an induced change in the differential proton-proton cross sections is expected as well. This expected change in the rapidities might lead to resolving the observed trend in y^*. Besides, the observed trend could as well signify a bias by the choice of the utilised PDF set, NNPDF 3.1, used in the generation of the samples. The trend might be mitigated by an alternative PDF set, or render a new derivation of PDFs necessary.

To estimate the effect of the models alternative predictions from other event generators like for instance Sherpa or Herwig, both implementing orthogonal models to Pythia, can be utilised and compared to the measured and predicted cross sections. However, the fact that a similar trend is observed in the non-perturbative corrections derived with the alternative event generator Herwig (see section 5.3.2.2) with different models implemented than Pythia suggests that the origin of the mismatch is found elsewhere.

Additionally, in the Herwig sample the PDF set CT10 [42] is utilised which is systematically different in the derivation than NNPDF 3.1. However, it cannot be excluded that NNPDF 3.1 and CT10 include the same systematic effects leading to the observed trend in the cross section predictions differentially in y^*. Consequently, more predictions

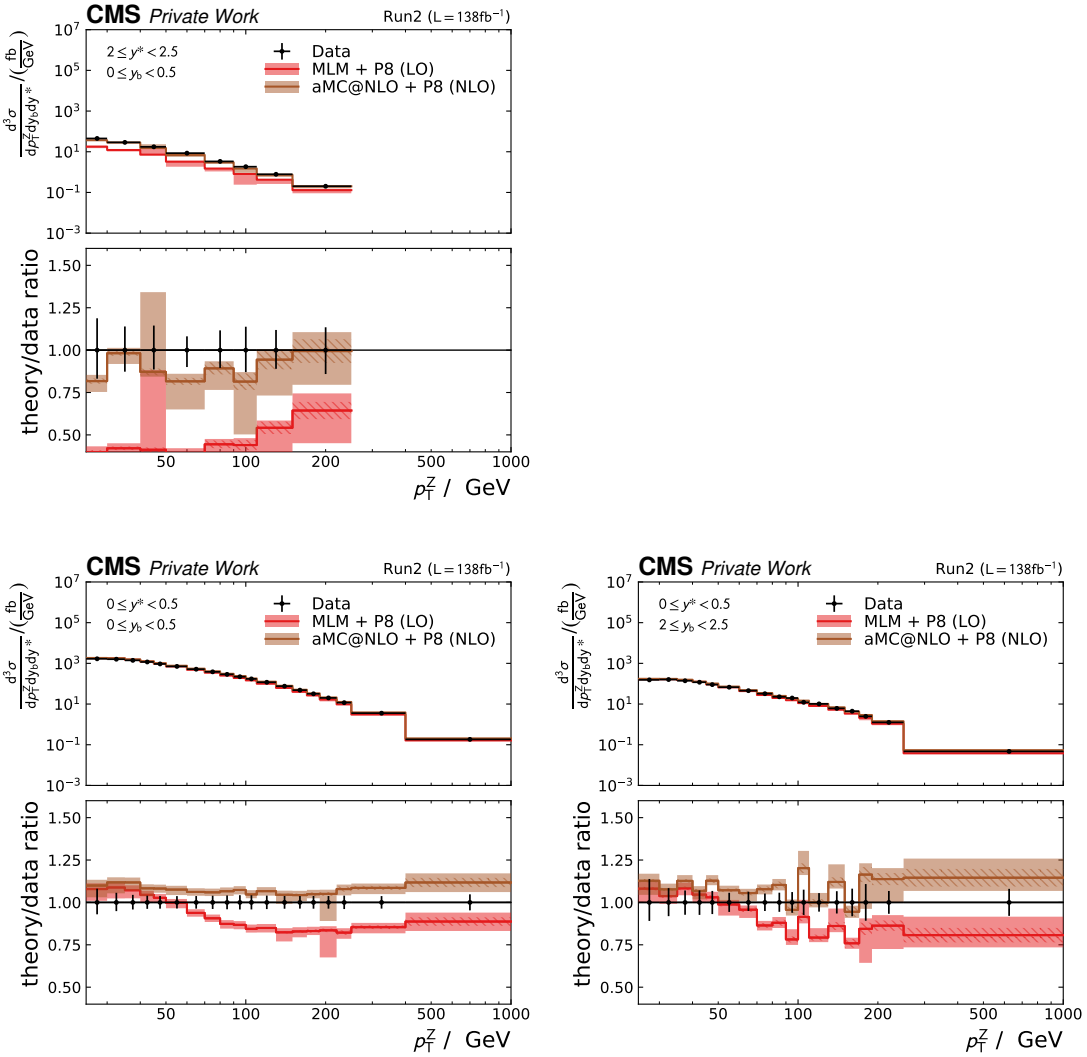


Figure 5.46: Measured cross sections corrected for detector effects (black) are compared to theoretical predictions at LO (red) and NLO (brown) accuracy in QCD for the central and two extreme bins in y_b-y^*. The uncertainties on the measured cross sections are the total uncertainties as defined in section 5.6. The uncertainties on the theoretical uncertainties include statistical uncertainties and parton shower uncertainties as defined in section 2.2.5.

are needed for comparing the influence of different PDF sets in the perturbative calculations in order to confirm or reject the hypothesis of a systematic bias or inaccuracies present in the utilised PDF sets which can be corrected in a new PDF fit including this measurement. Changes in the PDF have an impact on the rapidity distributions of the collision products due to their direct dependence on the momentum fractions in the colliding initial states which are governed by the PDFs. The differential cross sections measured in this thesis can provide valuable sensitivity for differentiating between such cases and therefore contribute to future PDF fits with higher precision.

For a study of higher order effects the corresponding state-of-the-art NNLO fixed-order predictions (see section 5.3.2) are needed. Higher order corrections in the perturbative calculation can improve not only the prediction of the inclusive cross section but also the predictions of differential cross sections for instance in y^*. Due to the size of the uncertainties for the predictions labelled with NLO no significant deviations are detected. However, the anticipated enhancement in prediction accuracy at NNLO accuracy in perturbative QCD potentially enables observations of significant deviations between predictions and measured cross sections. This can only be studied with the respective predictions available, which has not been the case at the time of finishing this thesis.

These investigations are left for future work.

CHAPTER 6

Modelling of Large-Scale Distributed Computing Systems

In this chapter, a method for the modelling and simulation of large scale distributed computing systems (LSDCS) is presented and the execution of workloads on such systems is evaluated. Although, the focus of this thesis is on the study of scientific computing workflows in the context of HEP collaborations and experiments, the generalization of the model to arbitrary contexts in which data is processed is argued. First, the conventions and common technologies and architectures in the HEP context are presented in section 6.1. Afterwards, the methods for designing efficient LSDCS and their feasibility and predictive error are discussed in section 6.2. Next, in section 6.3 simulation models for the description of LSDCS and their dynamic properties are examined and the chosen model for this work is described. Subsequently, a simulator tool based on the selected model is presented, and its applicability and, respectively, scalability for a small-scale system up to LSDCS with a high number of entities is evaluated. Furthermore, its predictions are calibrated and validated against data collected on a dedicated test system. In a final step, the tool's potential for the study of LSDCS is highlighted by an example inspired by computing systems designed for HEP.

6.1 Distributed Computing in the HEP Context

In the context of this thesis, distributed computing systems refer to systems of interacting components located on individual computers interconnected via a network. These components can communicate with each other by exchanging messages over the network of links and buses within a computer or between computers. In the following, the collection of computing components and connections organized in an exact architecture interconnected in a particular network layout will be referred to as a (computing) platform. Workloads running on such platforms will be characterized by their individual components, called jobs. Collections of jobs joined together by data flow and control dependencies, are called workflows.

In HEP, especially in the context of the Large Hadron Collider (LHC) experiment (see chap-

ter 3), distributed computing systems are crucial for the successful operation of the experiments and the physics program. The sheer amount of data recorded and simulated by the LHC collaborations, for instance CMS (see chapter 4), would be impossible to process on a single computer within the lifetime of an average human. Additionally, no single physical storage device exists, that could store this amount of data. To put this into perspective, in the year 2022 the CMS collaboration utilised approximately 1.94 billion CPU-hours in the European Grid Infrastructure (EGI) [86]. In the same year, the CMS collaboration required approximately 415 PB of tape storage in the WLCG [85]. For the whole lifetime of the LHC, CMS expects an increase by roughly a factor of ten in both CPU and tape requirements [137].

Fortunately, the data consists of records of independent collision events, which makes it possible to tackle the high data- and compute demands in a simple way: A single event is manageable in size ($\mathcal{O}(1\,\mathrm{MB})$ [65, 138]) and processing load. Additionally, events do not differ much from others measured or simulated by the same collaboration. Therefore, the data can be easily split into chunks – down to an atomic chunk size of a single event –, which can be processed and stored independent of each other. For convenience, events with similar features are arranged into a collection of related chunks, called a dataset. Those datasets can be distributed across many computing nodes that can each process a number of chunks independently. The resulting output data can either be first merged into larger chunks or directly stored individually at arbitrary storage nodes.

In a typical HEP analysis workflow, multiple processing steps are performed sequentially on the outputs of subsequent steps. These processing steps perform a data reduction or other transformation. Typically, the amount of information is reduced in several condensation steps, until in the end a manageable dataset size is reached, that can be processed by an analysis step and stored on a single or a few computers. This reduced data is then used for the final inference steps in order to generate a scientific result. Hence, the number of entities in the distributed system and the contained information decreases while progressing toward workflow completion. This hierarchical pattern is reflected in the structure of the distributed computing system utilised by the HEP community, the Worldwide LHC Computing Grid (WLCG).

6.1.1 Computing Resources, Sites, and Grid

In HEP and also in other fields, computing resources, when exceeding the number of a single computer, are typically bundled into collections of computers connected via a local network. This local system is typically called a computing site or centre. A platform of many distributed computing sites from multiple administrative domains interconnected through a wide-area network to reach a common goal is called a grid [139]. Such a grid is the WLCG introduced in section 6.1.1.1. Furthermore, utilised software and workload paradigms in the WLCG context are explained in section 6.1.2.

6.1.1.1 The Worldwide LHC Computing Grid

The Worldwide LHC Computing Grid (WLCG) [140, 141] is a global collaborative project to provide the storage and compute resources to the HEP experiments at the LHC and associated experiments. It combines several federations of data and computing sites connected via a worldwide network, following the idea of a distributed grid of computing resources [139]. Via junction points at the sites' edges, the individual sites are connected to the wide-area context of the grid network. Since the HEP collaborations operate via the WLCG, there exist overarching influences due to the inter-connections between sites. The WLCG was designed following the recommendation of the computing model Models of Networked Analysis at Regional Centres (MONARC) [142], which was derived with the help of the eponymous simulation framework MONARC [143].

The globally distributed computing centres are structured into hierarchical tiers. Raw data measured by the LHC experiments is saved on magnetic-tape storage and initially processed at the single Tier 0 centre located at CERN. From the Tier 0 the data is distributed to the Tier 1 centres, each located in a different region on the globe, for instance the German Tier 1 centre Grid Computing Centre Karlsruhe (GridKa) [144, 145]. They archive a subset of the precious data on their own tape storage and provide substantial computing power for reconstruction, simulation, user jobs, and collaboration wide analysis tasks. Additionally, they provide access to certain data to the Tier 2 sites for corresponding workflows, e.g. PU mixing files for MC event generation (see section 2.2), and storage for the simulation produced at the Tier 2 sites. Consequently, Tier 2 sites provide sufficient short-term online data storage and computing power for MC simulation, calibration studies and user analyses. At the time at which this thesis was written, there were 14 Tier 1 and 140 Tier 2 sites forming the WLCG [146].

A substantial fraction of user analyses are performed on research centres and university resources, commonly referred to as Tier 3 resources, which are individually managed and exclusively harnessed by local research groups or regional communities. Therefore, they are typically not in a common pool with the WLCG Tier 1 and Tier 2 resources and are therefore not directly utilisable by the collaborations. However, some are included into the common pool at the disposal of the managing institutes.

6.1.1.2 Third Party Resources

Third party resources consist of storage and compute capacities acquired under the premise to be managed or accessed only by regional or national communities (which includes also non-HEP communities), local groups or individuals. Typically, they are meant for sole usage by these clients and are therefore not necessarily included into the central resource and workflow management of the WLCG. They can reach sizes starting from a single desktop computer up to large supercomputers, consisting of hundreds or thousands of individual machines. In the context of the LHC experiments, however, they cannot operate completely independent of the WLCG, since they rely on access to the

data provided by the Tier 1 and Tier 2 sites, when running jobs with the aim to analyse this data. Consequently, to be usable for the LHC collaborations, the resources must provide compatible hardware and software.

Thanks to virtualization methods, for example containerization via Docker [147] and Singularity/Apptainer [148, 149], the HEP specific software environments can be provided, while the software can be distributed via network onto any system via container registries and CVMFS [150] without major overhead. However, as prerequisites, the security policies of the sites have to allow on the one hand the usage of these virtualization methods and on the other hand a network connection to the WLCG data servers and services.

Important examples of third party resources, which can provide a significant share to the available WLCG resource pools are, besides clusters affiliated to institutes associated to an LHC experiment, high-performance computing (HPC) clusters. Those HPC clusters – also referred to as supercomputers – are designed to solve single but very demanding and complex computation problems. As such, they typically combine hundreds to thousands of single computers within a low-latency network, which effectively allows the whole collection to behave as a single entity with a massive amount of parallel computational power and memory. This allows the simulation of large complex interconnected systems, e.g. climate or molecular models, which are too large to fit on a single machine. Since most of the typical HEP workflows involve compositions of single events, which can be trivially split or combined into arbitrary multiples of one event processed on a single CPU core, they do not benefit from the low-latency interconnections in HPC resources, but only from high bandwidths. However, the enormous amount of CPU cores and memory per core provided by such a centre can still be utilised. The HEP community profits from the fact that most queued workloads on HPC clusters typically are not or cannot be scheduled in such a way that the resources of the clusters are fully utilised due to the individual requirements of the workloads. Due to the flexibility in the HEP workloads, those free resources can be opportunistically backfilled by HEP jobs fitting the available capacity. If this capacity is needed again, HEP jobs are pre-empted. This allows the HEP community to use the HPC centre as a Tier 3 and supplement to Tier 2 and Tier 1 without disturbance of the main HPC workloads.

Similarly, cloud resources, for instance the ones provided by commercial cloud providers like Amazon Web Services [151], Google Cloud [152], and Microsoft Azure [153], can be booked as a service, which can be used to execute HEP workflows on demand [154]. The advantage of such cloud resources is that they can be swiftly spawned when needed, and are available in enormous amounts. Additionally, there is no need to directly and actively maintain the resources, since this is part of the service. However, the monetary costs for these services are usually too high to be cost-efficient for covering the basic demands of large-scale scientific workloads. Still, they can reach a reasonable regime, when they are offered with a discount due to science sponsorships by the companies, special run conditions like backfilling of the clusters, or used only temporarily during peak demands.

6.1.1.3 Heterogeneity and Complexity of the Grid

Although Tier 1 and Tier 2 sites are restricted in design of their computing platforms by the requirements due to their specific roles in the WLCG – for example grid protocols, instruction set architecture – each site is free to choose the types of worker and storage nodes. As a consequence, there exists heterogeneity in terms of the deployed hardware, which leads to non-homogenous CPU speeds and disk sizes, and of software used to deploy the services for the experiments. Additionally, due to the individual agreement between CERN and the local communities, the sites vary in size. Within a single site, due to operational and commissioning cycles rolling upgrades are performed every or every few years, the deployed hardware can differ very much in terms of CPU architecture, speed, number of cores, memory, disk sizes, link latencies and bandwidth. Adding Tier 3 sites into this system, where there are no restrictions on the deployed hardware and software, the heterogeneity of the whole extended WLCG system increases significantly.

Another level of heterogeneity is introduced by the network. The network adds a complex system of nodes like gateways, routers, and others interconnected by end-to-end network paths, which can be internal to a site or connect different sites. Only sites optionally connected to the LHC Open Network Environment (LHCONE) [155, 156] arrange on same terms. In general, there are no strict network interconnect guidelines or requirements imposed by the WLCG. Furthermore, large parts of the utilised network are beyond reach of the WLCG and are not part of LHCONE. Therefore, network components differ from each other in the global context of the WLCG but also within subsystems of the grid, for example in terms of latency and bandwidth.

Consequently, trying to predict the performance of an application workload on this system poses a non-trivial challenge. Assuming all information about the platform is known, neither finding the route a job's data would take through the network nor the actual machine running this job are easily accessible, since in the former case a complex network with many contributing devices is involved and in the latter a scheduler decision is made based on information about many potentially available resources. For a single job, this challenge still might be tractable with some effort, which would enable to find an analytical prediction. However, the amount of effort significantly increases when introducing several thousands of jobs competing for the same platform at any given time. Thus, finding a valid analytical prediction for a single job includes considering the effect of all the other jobs, because they share the same platform, in particular the network for which all jobs contend. Since in realistic workflows the jobs are not exact copies of each other, and the number of jobs is typically higher than the number of available slots of machines matching the job requirements, the cross-influence for a single job by the others is dependent on the exact configurations of the platform's software and hardware components at all times.

In conclusion, the state of the system as a whole at any given time, including the time-

dependent state of each job in the workflow as well as the time-dependent state of each platform components, contribute to the time-dependence of the system as a whole. Due to the scale of the platforms and workflows, which consequently lead to a high number of jobs and platform components, and due to their corresponding heterogeneity, performance prediction is a difficult challenge.

6.1.2 Software Infrastructures and Workloads

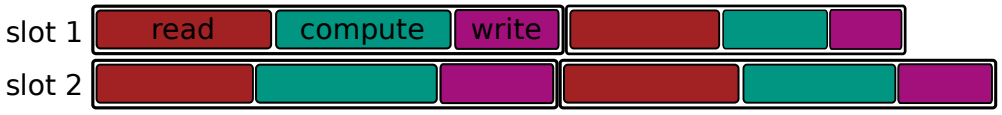
The processing of the recorded data measured by the experiments (see section 4.2.1) at the LHC, its reconstruction and analysis, and the collaboration wide MC simulations of the physics interactions and detector effects (see section 2.2) are composed of multiple workflows. Each workflow has different input-dependencies and compute requirements and needs to be periodically repeated with new data-taking, simulation or reprocessing campaigns. Often, each individual workflow task is composed of a chain of subtasks that can be individually run, but have to be run in the right sequence to obtain a reasonable result. The input data to be processed is distributed in advance among all grid sites. Therefore, to enhance efficient processing, jobs are preferably scheduled to sites where their input-data is present. The resulting increase in data locality reduces the network load and usually increases the efficiency due to higher transfer rates [90]. Due to the grid nature of the WLCG, each site operates primarily locally, but is at the same time part of a bigger network. This feature is relevant for services managing data and workflows across sites, which have to take the states of the local sites as well as the global system into account.

6.1.2.1 Workflow Management Systems

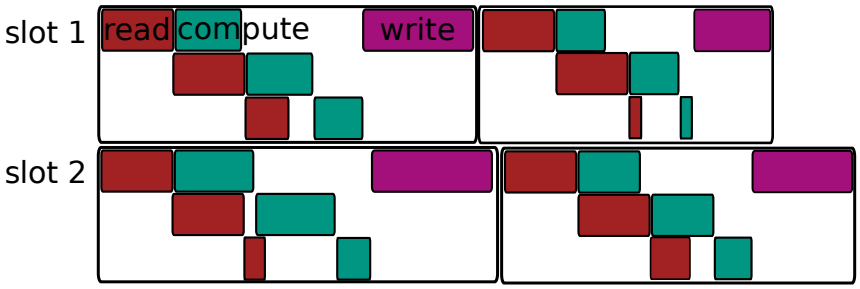
The WLCG poses a complex scheduling challenge for the collaborations, which is addressed by operating instances of a so-called Workflow Management System (WMS), for instance glideinWMS[157, 158] utilised by CMS [159, 160] or PanDA [161, 162] used by the ATLAS collaboration. Those systems manage the workloads consisting of jobs generated by the corresponding collaboration, identify suitable sites according to a collaboration-specific logic and book resources on selected sites. Next, the internal scheduling procedures of the sites match the booking requests to their individual compute resources, reserving the required resources. Once the resources are successfully booked and the WMS is informed about the new state, the actual jobs are distributed to the corresponding resources and processed. Similarly, the workflow management systems also partially steer the output locations for the jobs of each workflow. This is further elaborated below.

6.1.2.2 Job Schedulers

A common tool used for job scheduling at all tiers is HTCondor [163]. It is also integrated in glideinWMS [158] for the scheduling decision of the resource booking on the grid sites. HTCondor consists of four main types of interconnected services. Scheduler services keep track of the jobs queuing for a resource to run on. Start services run



(a) Sequential (batch) job



(b) Streaming job

Figure 6.1: While the execution time for sequential jobs 6.1a (also called batch jobs) consisting of single read, compute and write actions is given by the sum of times of each individual actions, the execution time for a streaming job 6.1b is composited of a more complex pattern. In a streaming job read and compute actions can be partially concurrently executed, since a subsequent read action can already start when the former has finished and the corresponding compute-action is run, which leads to a more compact execution pattern.

on execution machines and track their status. Both types of services advertise their information about the job and machine status to a central collector service, which gathers all information. The collected information about jobs that are waiting and machines that are idle is forwarded to a negotiator service, that matches a fitting machine to each single job respectively and advertises the free machine slot to the waiting job. HTCondor is primarily used in high-throughput computing context. For HPC applications, a prevalent scheduler tool is Slurm [164, 165].

6.1.2.3 Data Streaming and XRootD

The grid sites provide data access to the data stored on their grid storage via network over file transfer protocols, e.g. GridFTP [166] or XRootD [167], provided the client requesting the data is authorized. XRootD is of particular interest for data analysis tasks, since it provides the functionality to stream files block by block to and from the processing applications. In practice, this is usually relevant for streaming input-data from a storage server to a client running a data processing job. In contrast to a classical sequential job cycle, which copies all input-data to the local storage before reading it to memory, processes it and writes output-data to a remote storage at the end, input-file streaming allows the client to asynchronously perform the copy and read with the compute actions within the analysis tasks by rearranging the individual read and compute operations. Since read and compute actions use different hardware components on a resource, a subsequent read operation can already start, while the previous has finished

and the corresponding compute-action is executed. This can lead to shorter execution times of the jobs due to a higher concurrency of their individual actions. A comparison of the pipelining for sequential jobs (also referred to as batch jobs) and streaming jobs is illustrated in fig. 6.1.

Streaming is not the only functionality XRootD provides. One of the most prominent functions of the package is to provide file access without knowing the actual location of the respective file or its individual blocks. Block access is enabled by two components. First, the XRootD protocol enforces unique file and block identifiers. Files or blocks with the same identifier are considered to contain the same data. Second, a cluster management system enables to combine several data and other servers via network to a cluster and allows communication between its components. Inside this cluster, a server can run a redirector service, which redirects a request from a client for a file or block to a number of specified servers. These can again forward the request either to another redirector service or actual data server on the edge, which enables building directed graphs of server redirectors and data servers of arbitrary complexity. An example of such an infrastructure is represented by a directed acyclic graph in fig. 6.2.

When a data server is faced with a request, and it holds the requested file or block, a connection is established between the client and the server. Consequently, any file or block present on one of the data servers in the cluster can be accessed by raising a request to a central redirector, which is connected through a redirector chain to all servers. For example, the CMS collaboration, has built its XRootD infrastructure [168] as a directed acyclic graph with a tree structure, that is depicted in fig. 6.2. In the CMS hierarchy, there is a single global redirector at the top level and two central redirectors below for the American and Eurasian infrastructures, respectively.

6.1.2.4 Data Distribution and Access

The data collected and processed by the LHC-experiments (see chapter 4 for CMS) and the corresponding simulation (see section 2.2) are stored on long- and short-term storage on the grid. According to the design of the WLCG this data is distributed across all sites. Such a data distribution imposes challenges: On the one hand, data produced has to be assigned to the according sites storing that data. This poses a management challenge for distributing the individual datasets and replicas on the WLCG sites. The challenge gets more complex when instead of full datasets chunks of datasets need to be managed, as it is the case for short-term storage of CMS data. On the other hand, however, to be practically usable for analysis, the clients running the jobs analysing chunks of datasets need to be able to access the respective input data, which is distributed across many sites.

The first challenge is met by the WMS, which already gives a preference for at least the type of storage and site at which the jobs producing the data for each corresponding workflow should be placed. For CMS and ATLAS, the workflow management systems provide a filter logic for identification of a set of suitable sites and their storage systems.

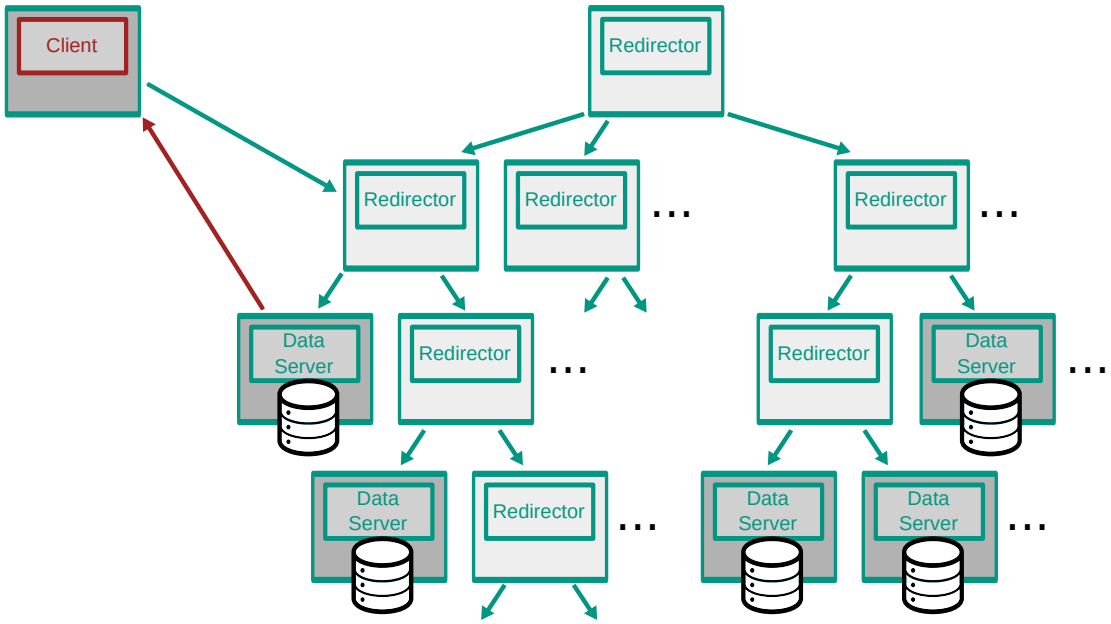


Figure 6.2: A client accessing data from an example server infrastructure organized as a directed acyclic graph of XRootD redirectors and data servers is shown. This example infrastructure of servers is structured hierarchically, resembling a pyramid. A client requesting a file present on at least one of the data servers can contact a redirector. From there, the request is forwarded down the redirector chain. Eventually, a data server holding the requested file is found and identified as a source. The possible paths for the requests are depicted as green arrows. Once a source for the requested data is found, a connection is established between the server holding the requested data and the client. Finally, data is transferred to the client (red arrow).

The actual selection of a specific site and storage system is performed by Rucio [169]. Rucio is a central manager service which keeps track of all available storage systems at sites and datasets with their relevant characteristics: the storage type and total available storage space on the one hand and on the other hand stored chunks of datasets, in this context referred to as blocks, with their size and locations. In CMS, Rucio accepts the filter logic from the WMS specifying the site and storage type preferences and randomly picks the site weighted by the remaining storage (total available storage space on that size minus the incremental size of the already occupied blocks) on that storage system [170]. Rucio also allows creating replicas on further sites, which can be issued by providing additional rules.

Since Rucio keeps track of all datasets and blocks, it can also be queried by a client trying to access a file in order to meet the second challenge. The client only needs to authenticate and provide a valid authorization for access to a file, since Rucio also keeps track of the block authority. In CMS the fallback and remote data access, and in particular data localization, is facilitated in practice by its XRootD hierarchy.

6.1.2.5 Data Caching

A cache is a data storage, that can provide previously placed data for a future request occurring in a finite amount of time. That data's origin can be a result of a computation or, more relevant for HEP, a copy of data placed elsewhere. If placed closer to the data requested on the network, the cache can provide the requested data faster than the origin.

In the context of distributed computing, in particular in the WLCG, data caches typically mean a volatile copy of data replicated with or without active management by an external service, for example Rucio. When managed, the data is placed on and evicted from the cache depending on the policies imposed by the external service. The external service can make its decisions based on information it gets from parts of or the whole system, including for example information about user behaviour, data access patterns, availability of storage servers or utilisation of parts of the network. An unmanaged cache is not provided with external knowledge about the rest of the system. Therefore, it can only make decisions in a locally restricted context based on its own state (total and occupied storage) and the data streams arriving or leaving (size of data and date of arrival or access). Depending on the available information, a locally defined logic governs the caching and eviction decisions.

Since data caches were not part of the original design of the WLCG, the concept of a data cache is not uniquely defined. In the context of this thesis, unless stated otherwise, data caches mean storages in the wide-area-network of the WLCG that provide volatile copies of data whose original replicas are permanently stored elsewhere. Additionally, the copy of the data is not centrally but locally managed by the cache itself.

Since the total amount of storage on a cache is limited, decisions on which data to cache

and, in case of a full occupancy, which data to evict from the cache have to be made. Identifying optimal, or even merely efficient, policies is not trivial, since they should take into account the total amount of data which can be cached, the characteristics of the data cached and the future access patterns of the cached data. The easiest non-trivial caching policy is to cache all data of a certain type, identified for example by file name or extension. For deciding which data has to be evicted in case of full occupancy of the cache, there are several established strategies. FIFO evicts the data which has been placed on the cache the least recently. LRU evicts the data which has been accessed the least recently. Depending on the data access patterns, randomized eviction strategies can also be effective. Additionally, versions of those decision algorithms weighted with the size of the data in question are valid variations. An overview of cache technologies and eviction policies is given in [171].

The functionality for defining a service running a local cache is provided by XRootD [167]. XRootD allows starting a so-called proxy storage service. When a proxy is contacted by a client, the proxy will forward the request to another server. Once a data stream is started, it will be routed to the proxy server. At the same time, the transferred data will be cached in memory of the proxy and the data from there will be provided to the client. Optionally, the proxy can also be configured as a disk caching proxy, which means that the transferred data or parts of it is cached on a local disk instead. The next time a request for the same data is raised to the proxy, the data will be provided from the cache. An example sketch for a data access via a caching proxy is depicted in fig. 6.3. The default cache directive in XRootD is to cache all incoming data streams. The default policy for eviction of data on the cache is to remove least recently used data.

6.1.2.6 Dynamic Resource Provisioning

For the dynamic integration of resources there are several tools available, for instance Roced [172, 173], COBaLD &TARDIS [174–176], cloudscheduler [177] or HEPCloud [178]. The respective resources they manage might only be accessible to a group of clients for a short finite amount of time like compute nodes on an HPC centre or a Cloud (see section 6.1.1.2). When demand for and supply of suitable resources is available, these tools book available resources on eligible sites and temporarily integrate these into the pool of available resources. The detailed matching procedure of these tools differs, but they have in common that each defines some metric for supply and demand and tries to find the best matches between those. COBaLD &TARDIS and cloudscheduler try to optimize the utilisation of the opportunistic resources by defining a suitability metric for the queuing tasks onto the resources. This metric contains additional information about the resources and the jobs, which can be used to find optimal matches. It can be utilised to avoid booking more resources than the demand involves or resources, which cannot be efficiently utilised by the queueing jobs. This leads to a more efficient usage of the advertized resources.

Since those metrics are typically time-dependent, at least two approaches exist to max-

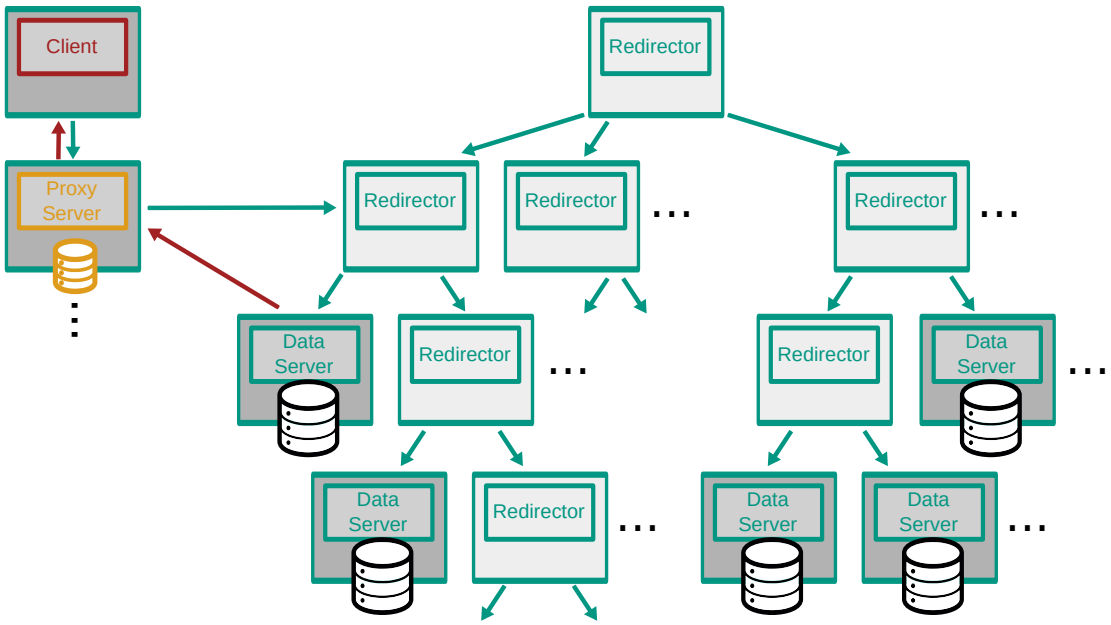


Figure 6.3: A client accessing data from an example server infrastructure as shown in fig. 6.2 via a proxy data cache is shown. Once a source for the requested data is found, a connection is established between the server and the proxy, which then provides the data to the client (red arrows). The next request for the same data by a client via the proxy will be served directly by the proxy, if the data has not been evicted in between those two requests.

imise the utilisation. On the one hand, predicting the metrics typically is a challenging endeavour, given the heterogeneity and sheer number of the workloads and eligible resources. The resulting complexity of the interconnected system of platform components, jobs interacting with this platform and each other, makes it unfeasible to achieve an appropriate forecast for the necessary metrics accurate enough to be beneficial. On the other hand, dynamically reacting to the contemporary conditions of the system is more viable. Although this approach will only give snapshots of the system's status, which are outdated shortly after obtaining them, this turns out to be good enough to provide a benefit, as has been shown on several occasions, for example for COBALD &TARDIS [90, 179–181] and cloudscheduler [177, 182–184].

6.2 Design of Large-Scale Distributed Computing Systems

Generally, large scale distributed computing systems (LSDCS) consist of an interconnected network of interacting computing components located on individual computers. This allows these computers to combine their individual capacities efficiently sharing a workload towards a common goal, which would not be feasible for a single component to achieve in limited time. In order to design an operative computing infrastructure for the execution of work defined by the users of this infrastructure, three crucial components need to be considered. First, the architecture of the physical infrastructure itself providing the hardware platform for executing computing applications, which characteristics and demands of its workloads defined by the users make up the second component. Last, the distribution of the workloads on the infrastructure, which inevitably connects the first two components. The complexity of such large scale distributed computing systems (LSDCS) as well as their size makes the design of efficient systems a challenging task.

6.2.1 Complexity of LSDCS

Large scale distributed computing systems (LSDCS), i.e. the WLCG (see section 6.1.1.1), consist of many entities of different sizes and types. Although the WLCG shows a hierarchical ordering of its consisting sites into four tiers, the sites' features vary widely. Although there are minimal requirements, it is on the sites' administrators to decide on the specifics. For example, the number of worker nodes as well as their explicit features, e.g. the number of and the specific CPUs, the amount of RAM, the size and bandwidth of their local scratch storage and their connection to the LAN are not defined by the requirements of the WLCG. Also, the sites are in the same manner free to pledge the size and bandwidth of the provided storage and internal composition thereof. Furthermore, the design of their LAN and its characteristics, e.g. bandwidth, as well as its outbound connection to the WAN are each site's responsibility.

The demands on a LSDCS, like the WLCG, manifesting in the workloads and jobs sent by its users to be executed on its hardware are manifold. As described in section 6.1.1.1, the tasks defined by the LHC collaborations on a site depend on its placement in the hierarchical structure. However, for each of the tiers the collaborations have a range of

tasks, which vary in their characteristics and requirements and are subject to dynamic changes. They also differ widely between the collaborations. Additionally, individual users can also use the WLCG. The workloads those users start on the infrastructure are as diverse as the users and are typically not subject to any restrictions or requirements imposed by the WLCG or the users' collaborations. As a consequence, the individual building blocks of the tasks and workloads defined by the users of the LSDCS, the jobs, vary significantly. They differ even more so when taking into account that for each workload or task the corresponding jobs' characteristics can vary, for example due to wide ranges of input file sizes.

In order to distribute the jobs to the infrastructure that will process them, job schedulers (see section 6.1.2.2) are crucial to ensure a successful execution. Based on the jobs' estimated requirements – the true characteristics and requirements of a job are often not know initially – an appropriate section of the platform has to be identified based on the advertized characteristics provided by the components in question. This matching process can be further optimized by taking the (current) state of the machines, e.g. the current occupation of its resources, into account. However, taking more information of the system into account for the scheduling decision increases the complexity significantly. Not only by an increase in the complexity of the scheduler's logic, but also since there has to be more information provided by the users about the estimated characteristics for their jobs and by the machines' operators about the state and features, which can both be subject to errors and uncertainties.

With the scheduler connecting the workloads and the infrastructure running these workloads, a coupled system of job and machine entities interacting with each other emerges. Depending on the scheduler decision, jobs can share parts of the same machines or possibly run across more than one single machine (although the latter is not considered in this thesis). This creates interdependencies between jobs. Additionally, since machines might run several jobs, their execution might lead to contention on parts of the network. This connects jobs even when they are not located at the same site. Therefore, the execution of a single job cannot be analysed in isolation. That is, approximations that, for example, regard jobs dependent on only locally close machines need to be carefully evaluated, since the jobs executing on further afar might have a non-negligible influence on the local execution. Overall, in general, the whole system with all its complexity has to be considered when the dynamics of applications running on large scale distributed computing systems (LSDCS) are studied.

6.2.2 Testbeds versus Models

In order to study LSDCS typically two options are suggested: First, the construction of dedicated testbeds. Second, the design of empirical mathematical models that aim to reproduce the behaviour of real systems realistically.

6.2.2.1 Testbeds

Building up infrastructures of real machines and running while monitoring real applications on those systems is an obvious approach for ensuring that performance studies are performed with no bias. In order to investigate different infrastructure designs, the dynamics of the running applications can be directly monitored and their execution on different testbeds can be compared. Additionally, no in-depth knowledge about the systems' hardware and software components is needed, since the only requirement is that the system is operational from a usability perspective. Unfortunately, however, the costs of building up a testbed puts tight limits on this approach. The required hardware as well as the commissioning of the test infrastructure imply possibly large monetary costs. Furthermore, when building large systems that consist of many hardware components or many systems of different design requires a lot of labour, quickly making this approach unfeasible.

Already, building up a single testbed sufficiently large to capture the realistic behaviour of workloads running on the WLCG is out of reach, since it would require a twin of a global infrastructure with $\mathcal{O}(10^5)$ components. As an alternative, testbeds that are representative of isolated subsystems of the WLCG of manageable size can be built-up and used for testing a specific design. However, as discussed in section 6.2.1, the obtained results will not coincide with the full system's, since the non-negligible external influence is eliminated. Another option can be to build a surrogate architecture of the full one, which is scaled down in size while keeping the complexity of the original, thus conserving the realistic dynamics. Unfortunately, it is a priori unclear how this is achieved and can only be validated by comparing to the full-scale system. Also, since the system is built up of individually countable components, the down-scaling is limited when one of the components becomes a single unit. Scaling below this threshold is obviously not possible for discrete entities. When the original system is large, this might not be sufficient for making the surrogate testbed small enough to be feasible and yet representative.

6.2.2.2 Models

The alternative to building testbeds is to model the behaviour of a LSDCS and to use this model for predicting the performance of hypothetical applications or architectures. Those models can be based on first principles or on empirical knowledge about the systems. There are two popular types of models.

On one hand, one can develop analytical models that formulate the dynamics of a system in terms of a limited number of mathematical equations. Typically, this means a set of differential equations or stochastic relations. Solving these with a sufficient set of boundary conditions lead to predictions for the dynamics of the system under surveillance. Unfortunately, for a LSDCS, because of the scale of the systems, large numbers of components are interfering with each other, which lead to many strong couplings between individual components and therefore the equations in the analytical model.

Moreover, the interference is in general not linear. As a consequence, deriving solutions for analytical models for LSDCS is often not tractable, without simplifying assumptions, which might deteriorate the realism or limit the scope in which this specific model can be trusted, see e.g. [185, 186].

On the other hand, one can develop simulation models, e.g. discrete-event simulators, that are composed of several interconnected modules aiming to model specific components of real systems. For each component, the model can consider characteristics and properties specific to its type. Those components exchange information, mimicking the behaviour of real systems. Indeed, those components are themselves typically governed by or composed of analytical models at their heart. The information exchange itself is also characterized by a model and can differ depending on the types of the exchanging modules. This leads to a set of model parameters for the characterization of the modules as well as the information exchange. The values of those parameters are a priori undetermined. Therefore, it is crucial to tune the parameter values such that the model can lead to realistic predictions. This is also called calibration of the model. Of course, there might be no set of parameter values that achieves the desirable behaviour. This hints to a bias or missing component in (parts of) the model and can be approached by revisiting certain assumptions or designs made while constructing it. Once all parameter values are determined, predictions can be numerically obtained by executing all the events in the right order and keeping track of the simulator's state with each step.

Building any model that is able to capture the real world systems realistically requires in-depth knowledge about the functional principles of the considered components. For example, modelling network transfers over TCP/IP will only give realistic results when taking the specifics of those protocols into account [187]. Oversimplification in contrast, will lead to predictions, which do not reproduce the dynamics of real systems. This however makes a validation of those models crucial in order to identify possible modelling biases, deteriorating the accuracy of the model. Hence, the predictions made by the model need to be compared to data gathered from real world systems, requiring the monitoring of these systems and analysis of the gathered and generated data.

When building a model the aim is to sufficiently capture the complexity and variety of real systems in order to be able to obtain realistic predictions. However, with increasing complexity of the model also the effort for obtaining predictions as well as the number of terms or entities to keep track of increases. Subsequently, this increases the time and used memory to obtain the results. Reducing the complexity for saving time and memory, however, might harm the accuracy of the model. This trade-off between accuracy and computational complexity of the model has to be considered when coping with LSDCS.

Both analytical and simulation models have been subject to research with the aim of optimizing the efficiency, or identifying efficient configurations of distributed computing systems. However, most of the time those studies aim for improving specific portions of

the overall system. Therefore, the models used only cover confined regions of specific LSDCS in mind or cope with only a limited range of protocols, software technologies and methods used in such systems to the benefit of reducing the computational complexity of the model. Restricting the model to a scope that is aimed at a specific experiment or scientific question in mind is per se not problematic. As a consequence of the restriction, however, there is a possibility for a bias introduced by the isolation of the context from the bigger scope. This bias, if noticed at all, is often argued to be negligible based on subjective reasons or assumed to be negligible as a precondition of the specific study. A minimal requirement to avoid a significant bias in the model is therefore the validation of the predictions with real world data.

6.2.3 Example Models for LSDCS

In [188] a model based on [185] for the data flow in parallel computers, i.e. HPC where communications between processors are naturally included (see section 6.1.1.2), is derived. They present a purely analytical solution for a system of partial differential equations describing the full dynamics of any hypothetical parallel computer system. However, the model is not validated and the question about the applicability to real HPC systems is not answered. Also, this model is strictly limited to parallel computers and can only describe the data flows.

There also exist a full class of stochastic models for evaluating the performance of distributed computing systems, e.g. stochastic petri nets [189, 190] or process algebras [191]. These models however, focus on the resource part of computing systems only and do not allow to include the influence introduced by the applications running on those systems. Therefore, the stochastic models are extended to support transformations of the resource model in order to describe the influence introduced by the running applications on a real system, e.g. [192, 193]. Further extensions allow communications between multiple software components, modelling the applications, and implement configurable functions for timely dependencies of their parameters, e.g. [194, 195]. This allows taking dynamically changing applications into account.

These last models are very close to simulations, since they already implement a notion of logically distinguished but interconnected models aimed at describing a real system in its entirety. In this thesis, the simulation model means the ensemble of all the combined individual models interacting with each other, as described in section 6.2.2.2. However, those individual models can also be of the same type. Such a simulator with multiple components of the same type described by the same underlying model is for example ns-3[196]. It simulates transfers over arbitrary networks via TCP/IP on an individual packet level. Although validated to describe real systems with high accuracy, the detailed simulation of many agents leads to long execution times, which makes it unpractical for the use in simulations of LSDCS [197, 198]. Nonetheless, ns-3 is used as a model for network simulation in many other simulators, since it is able to capture most properties of real networks and therefore shows a high accuracy.

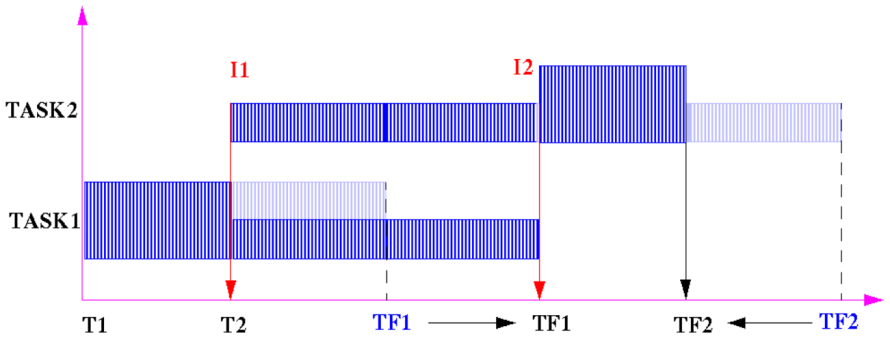


Figure 6.4: Interrupt-model of the MONARC toolset handling the shares of resources between concurrently running tasks, taken from [143]. When a task starts a share on the used resource is assigned based on its priority or equal shares when none. This share is considered constant until a new task arrives, or a task is finished. These events interrupt the running tasks and new shares are assigned.

The simulator MONARC [143] was crucial in the HEP community, since it lead to the initial design of the WLCG (see section 6.1.1.1). It considers three major components, which build up the real system architecture: Data is modelled as containers, which abstract a sequentially ordered collection of data objects, as the atomic unit. They are stored on mass storage server entities with several storage management policies implemented. Access to the data is modelled via database servers with response times depending on the data parameters and hardware load at time of access. Computation is modelled as strict data processing tasks possibly sharing the same resources of CPU, memory and I/O. The resources are assigned based on an assigned priority or equal shares otherwise. The shares are updated, when a task starts or finishes on a resource, defining an event. In between interrupting events the shares are assumed as constant. A visualisation of the resource share concept is depicted in fig. 6.4. The network over which all I/O is streamed is modelled without specifying a network topology beyond links for each LAN and WAN component inside and in between regional centres described below. Instead, time dependent-functions for each link that describe the effective bandwidth on that link need to be defined by the user. As a consequence, the effects of packet loss, overheads, outside traffic and specific protocol features, e.g. round trip time unfairness in TCP/IP, need to be specifically estimated by the user. Shares by individual tasks are assigned according to the same interrupt model described above. When multiple links contribute to an I/O task the minimum bandwidth is used to determine the progress between two events. Activities by (groups of) users of the simulated system are modelled by user-defined job submission patterns. Each activity is assigned to a single regional centre (see below). The simulated platform of resources is modelled as a number of regional centres connected by WAN links. Each regional centre itself consists of a number of data servers and processing nodes and an optional mass storage server connected by a LAN link. For each regional centre a specific scheduling policy can be assigned for manag-

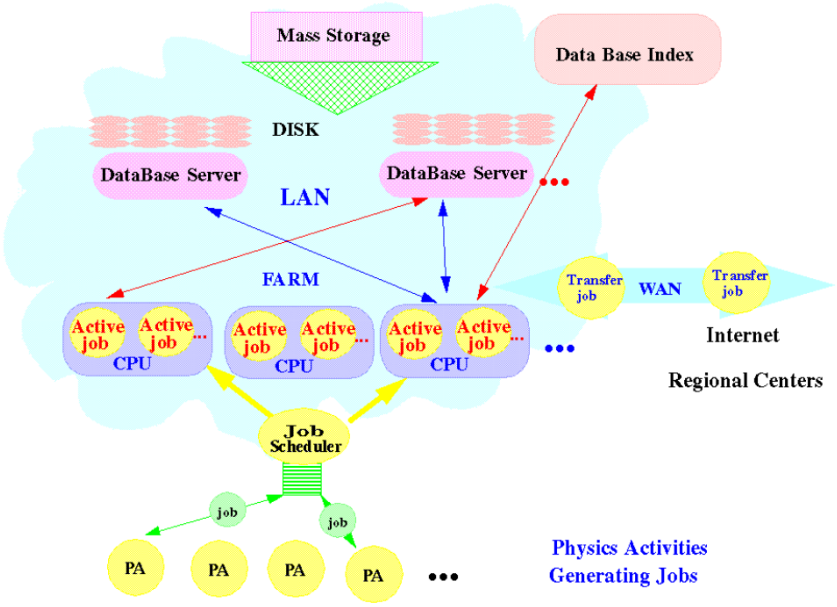


Figure 6.5: Schematic of the regional centre model in MONARC, taken from [143]. A regional centre consists of data servers, processing nodes and an optional mass storage server connected by a LAN link. Each regional centre has its own activities assigned and an implemented scheduler managing those. Multiple regional centres can be connected by WAN links in between them.

ing its activities. A schematic depiction of the regional centre model is shown in fig. 6.5.

Although MONARC has played an important role in HEP computing, the original MONARC [143] and its successor MONARC2 [199] have been discontinued.

More recent development projects on simulators for LSDCS are for example OptorSim [200] / GroudSim [201], which originated in the HEP community and GridSim [202] / CloudSim [203] and SIMGRID [204] with both serving as a base of research for hundreds of publications each, see [205, 206]. Whereas GridSim / CloudSim and OptorSim showed issues in reproducing reality in validation studies especially for network simulation, SIMGRID overcomes these issues and is able to produce realistic predictions [187]. Therefore, the models implemented in SIMGRID are chosen as a basis for the simulator presented in this thesis. They are discussed in detail in section 6.3.1.

6.3 Simulation of Large Scale Distributed Computing Systems

In this section, the underlying models utilised in the implementation of a dedicated simulator tool are described in section 6.3.1. Extensions to these models and HEP specific

adaptations are combined to compose a simulator tool that is presented in section 6.3.2. The code for this tool is published in [207]. The calibration and validation of this tool is discussed and demonstrated in section 6.3.3. Its computational complexity is studied in section 6.3.4. Finally, an application of the tool in a study of a hypothetical LSDCS is presented in section 6.3.5 showcasing the potential of simulation.

The presented simulator, parts of the presented calibration and validation procedure and parts of the studies performed with this tool have been originally published in [208].

6.3.1 Simulation Models

Simulation models and their implementations in simulators have widespread use for theoretic experimentation and design of performant distributed computing applications and architectures. They are compound out of ensembles of interconnected components, each governed by its own model, see section 6.2.2.2. As such, each sub-model as well as their interaction have to be defined and validated. Out of the presented models in section 6.2.2.2 only SimGrid [204] satisfies the requirements of being widespread and usable, validated and accurate and scalable and expressive for complex architectures encountered in distributed computing at a scale encountered in the WLCG. Therefore, SimGrid and Wrench [209], which provides high-level abstractions built on top of SimGrid, were chosen as a base for the simulator presented in this work. In the following, both tools and their underlying models and assumptions are described in detail, following the descriptions outlined in [204, 209, 210].

6.3.1.1 SimGrid

SimGrid is a library of low-level simulation abstractions implemented in C++. As such, it is a framework for the development of simulators for distributed computing architectures by using APIs available in C/C++, Java and Python. Under the hood, it implements a range of models for the simulation of hardware components, called resources, which run distributed applications that consist of interdependent activities.

Engine – The execution of a SimGrid simulator consists in simulating the execution of user-defined actors, which spawn activities that use simulated hardware resources. These activities can be used for inter-actor synchronization as well as for simulating consumption of resource capacities for performing the simulated application's work, for instance computations on compute, data read and write operations on storage and communications on network resources. Actors can dynamically create other actors, and all actors are managed by a special actor called the maestro. This maestro is akin to an operating system and is in charge of scheduling all other actors and keeping track of the simulation clock. In a scheduling round, the maestro passes control to all actors that are not blocked due to a dependency on an activity that has not yet completed. These actors execute their user-defined activities and return intermediate status signals until all of them become blocked. Once all actors are blocked on pending activities, the

maestro invokes the internal simulation models to determine the activity completed the earliest. The simulation time is advanced to this time, and the remaining amounts of work for all remaining activities are updated. Actors that were blocked on activities that have completed are unblocked and the next scheduling round is started.

Resource Usage Model – SIMGRID uses a unified analytical model for determining of an activity's progress independently of the type of resource used. The activities are characterized by a total amount and a remaining amount of work to accomplish on assigned resources. Since there can be multiple activities claiming the same resources, the capacities C_r provided by resources r have to be shared among the set of all activities $\mathcal{A}$. The share ρ_a for an activity $a \in \mathcal{A}$, which determines the future progress of the activity, is determined by solving the constrained optimization problem

$$\max \left[\min_{a \in \mathcal{A}} (\rho_a) \right] \tag{6.1}$$

under the constraints

$$\sum_{a \in \mathcal{A}} \rho_a \leq C_r \tag{6.2}$$

for all r. Once the ρ_a are determined for a time stamp t_i, the time is advanced to the time stamp t_{i+1} at which the first activity completes or a new activity starts. As a result, the remaining work for each activity $a \in \mathcal{A}$ is decremented by $\rho_a (t_{i+1} - t_i)$.

The optimization target is chosen in this way, in order to implement Max-Min fairness [211]. The idea behind this is, that increasing the allocation of any ρ_a would require decreasing the allocation of a less favoured one, while accounting for the fact that certain activities involving multiple resources can utilise more share than others.

CPU Model – Activities demanding CPU resources define their work in terms of compute costs. Resources representing CPUs or CPU cores are characterized by a (time-dependent) CPU speed. Consequently, the resource usage model above introduces simple analytically determined compute delays due to compute activities. As an extension, SIMGRID allows weighting the ρ_a in eqs. (6.1) and (6.2) with a weight determined based on a user-defined compute priority per activity.

As an edge-case, for a set of unweighted compute activities occupying only a single resource each, eqs. (6.1) and (6.2) result in a fair share for each activity on the same resource.

This model does not take the internal structure of different CPU architectures with internal buses, caches etc. into account. Instead, the CPU model abstracts all those internal features into a single characteristic CPU speed. Nonetheless, for most experiments the simplifications are sufficient. However, if required, the platform description of SIMGRID along with activity characterizations can be harnessed to approximate a representation of the internal structure of CPUs (see below).

Storage Model – Data access times are modelled in SIMGRID by a combination of seek time and transfer time. The transfer time is determined utilising the optimization procedure in eqs. (6.1) and (6.2), where the work to be done corresponds to the amount of data to be transferred, and the resource capacity is expressed in terms of a data transfer rate. The seek time is a parameter of the specific resource and is added initially when advancing the simulation clock.

Like the CPU model, this model also simplifies the behaviour of real storage systems. File system effects, data locality, caching and buffers which drive the performance of real storage systems are abstracted into two single characteristic parameters. Yet, those simplifications are in general sufficient for most utilisation in the context of distributed computing. Some neglected effects, however, e.g. storage buffers can be addressed by adjusting the activity traces to represent a more realistic structure (see section 6.3.1.2).

Network Model – Since packet-level simulation models that capture most of the details of real network transfers over TCP/IP, e.g. ns-3[196], scale poorly with the size of distributed applications (see above), ns-3 is intended in SIMGRID as an alternative. By default, SIMGRID implements an analytical network model as an approximation of the packet-level simulation based on the resource usage model, see eqs. (6.1) and (6.2).

However, since the network in the context of LSDCS is the central component which connects all the local entities, i.e. CPU and storage resources, makes putting special emphasis on the validation of any simplified network model necessary. Indeed, many other simulation models for LSDCS with simplified network models fail to do so [187] which casts doubt on their validity or restricts them to a limited application scope. Yet, it was shown in [187] that the approximate network model implemented in SIMGRID is able to simulate network transfers with high accuracy.

Network transfers in SIMGRID are approximated as a continuous flow of data, instead of individual packets. Between two events, this flow is fully characterized by a constant data rate, since it is assumed that all flows through the network are laminar. The work to be executed in a data transfer activity is given by the amount of data, which needs to be transferred. The resources executing the work are network links, which are assigned a bandwidth and a latency. Since there can be many links contributing to a transfer of data, the data rate at a specific time for a specific transfer is a result of the interaction with other concurrent data flows and the network topology.

Unfortunately, network protocols like TCP/IP do not follow Max-Min fairness [212]. Therefore, for the determination of the bandwidth shares eqs. (6.1) and (6.2) have to be adjusted. In order to be able to account for the RTT unfairness of TCP [213] and throughput degradation due to reverse traffic [214], eq. (6.2) is adjusted [187]. With this adjustment, the assigned data rates ρ_a can be determined accurately for most scenarios.

Furthermore, the execution time of a transfer

$$T_a = \alpha \ell_a + \frac{V}{\beta \rho_a} \tag{6.3}$$

is adjusted by two empirical parameters α and β, which are tuned to a specific transfer protocol. Here, α denotes to a scale parameter which adjusts the latency ℓ_a and β scales the assigned data rate ρ_a.

For the sake of completion, there remain network transfer scenarios that cannot be accurately described by this flow model. These issues arise because the laminar flow assumption is violated and the discrete nature of the transfers becomes relevant. However, those correspond to situations with data sizes smaller than 100 KiB or high contentions on links with low capacities. Fortunately, these situations are irrelevant for HEP application where the data sizes are much larger and high capacity links are utilised.

Platform Description – A simulated hardware platform in SIMGRID, which represents the architecture of a computing system, consists of hosts and routers connected by a network of links. Routers in SIMGRID can be seen as minimal hosts in the sense that they only provide a junction for links. As such, they are not instantiated with a model for further functionality and are only relevant for the routing of network transfers. Hosts also connect to links, but additionally contain storage and CPUs subject to the models described above. Consequently, hosts and links represent the simulated resources for CPU, storage and network and need to be characterized accordingly by the user setting the respective parameter values.

Additionally, the topology of the interconnections has to be defined. This is achieved by a definition of the allowed routes – the chain of links data transfers between two hosts are allowed to traverse. These routes can be explicitly given for each host pair combination, which requires a lot of effort from the user. Alternatively, the routes can be given only for directly connected hosts and routers building up a topology graph. Full routes for explicit transfers between two endpoints, which are not explicitly stated, can afterwards be resolved from this information by determining the shortest path on the topology graph. The figure of merit for the shortest path determination can be e.g. the number of links to traverse or the path with the smallest latency. The former would be constant for a given platform, while the latter might change with each event and would need to be recomputed every time.

In both routing options, for n hosts and routers the number of possible routes grows at $\mathcal{O}(n^2)$. In order to decrease this computational complexity, the typically hierarchical structure of real network infrastructures can be exploited. An example hypothetical infrastructure as it might be used in HEP for processing is depicted in fig. 6.6. Generally, real-world networks can be viewed as consisting of LANs interconnected by a WAN or even multiple convolutions of this concept. In this case, the topological graph can be simplified by grouping clusters of hosts and routers into their corresponding local zones.

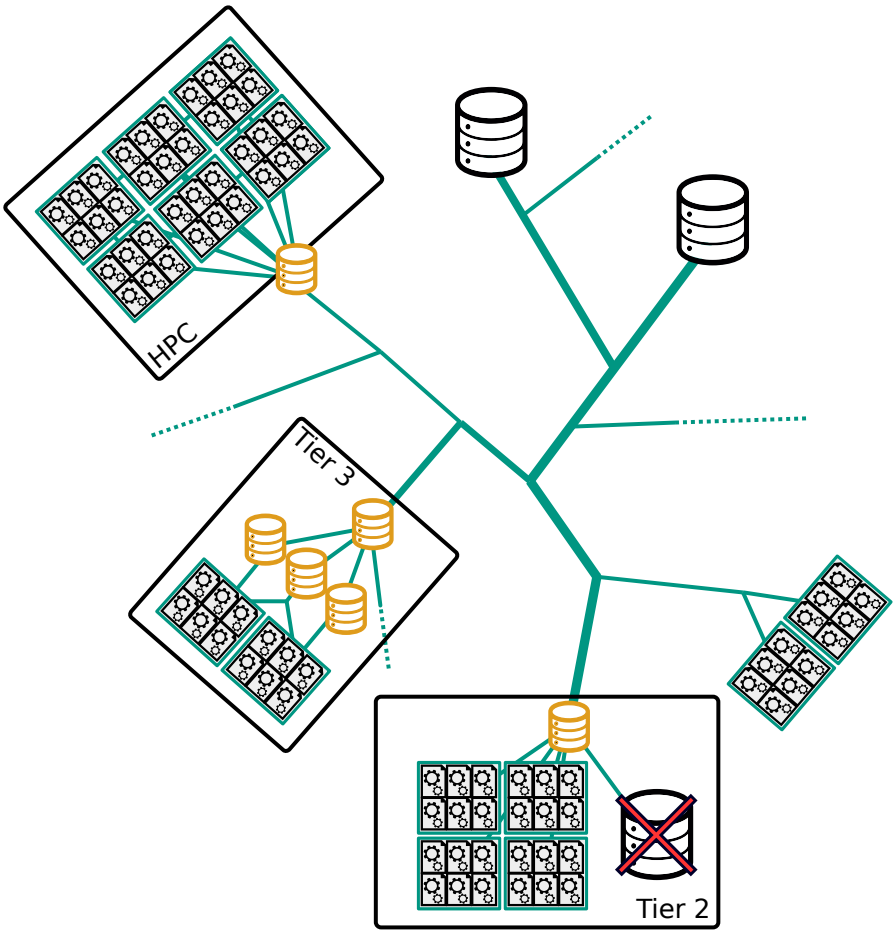


Figure 6.6: Hypothetical example for a part of a computing architecture used in HEP data processing. It consists of a network of worker nodes (black gears) or clusters of those and servers functioning as permanent and volatile storage or cache (black and orange stacked disks) interconnected by links (green lines). Parts of the network can be grouped according to the site they belong to into local zones, e.g. an HPC cluster, a Tier 2 and a Tier 3 site. These zones can be viewed as independent LANs which are connected to the global WAN.

Those zones define their own routing scheme and are connected to other zones via a zone gateway. This divide-and-conquer approach simplifies the routing significantly, since the routing can first be determined locally and the result passed on to the enclosing zone afterwards.

The resulting notion of a platform leads to a lot of versatility in setting the scenario to simulate. Moreover, it allows the user to set the level of accuracy for his simulation. For example, the abstract notion of a host could be used to configure a simulation where the platform is described with each worker node of a reference real-world architecture assigned a host and they are linked according to the real-world network topology. This allows to model a computing architecture in great detail, which will probably also increase the accuracy of the simulation. However, at the same time this would also be computationally more expensive than the following alternative. By contrast, if the internal structure of parts of the architecture are not relevant, e.g. the structure of a cluster of worker nodes, the characteristics of this part can be condensed into a single host. In that case, the effective characteristics of that compound host have to be estimated from the internal structure, e.g. the number of cores obtained from the sum of the cores of all contained worker nodes of the cluster. It has been shown in [198] that this versatility can be utilised to create models for a wide range of different architectures.

As a mere thought experiment, this concept can be escalated down to the scale of a single computer. In this context, a CPU could be modelled as a host with only CPU capacity, memory, and storage could be modelled as two separate hosts with no relevant CPU and only "disk" functionality with accordingly largely different I/O bandwidths. These hosts would be connected by links representing the internal machine buses. Certainly, the application which is tested in the simulation to run on this platform would need to utilise it accordingly in order to get an accurate result. However, to the best of my knowledge this has never been validated and remains a pure illustration.

Platforms can be either defined programmatically, using the API provided or in XML. Examples of platforms defined in XML are given in appendices B.2.1 and B.2.3.

6.3.1.2 Wrench

Simgrid provides the basic functionality for defining an accurate model for any kind of LSDCS. However, the flexibility in its models comes with a trade-off. It is labor-intensive to build representations of complex real-world systems and the corresponding applications, since both have to be written from scratch. These issues are addressed by Wrench [210]. Wrench provides high-level abstractions based on the Simgrid models which can be used as convenient building blocks for implementing a simulator of a complex system. These building blocks are structured in several layers.

First, low-level hardware and software stacks are implemented, which interact directly with the models introduced by Simgrid. The second layer consists of services abstracting

compute and storage resources, and network-monitoring, data-registry and energy-meter stacks which interact closely with the platform. An API provides the functionality for programming those services. For each of these categories of services WRENCH provides already multiple implementations, which correspond to use-cases relevant for LSDCS simulation, e.g. bare-metal, batch and cloud compute services. The third layer consists of controllers, which manipulate and steer the services. The example given in [210] would be a workflow management service, which gathers the information about jobs to run and matches and schedules them to the available resources depending on the specifications of the resource services and the job dependencies. Another example could be a controller which changes the state or parameters of a service dependent on predefined conditions. The last layer is the one closest to the user. Its API contains the functionality for instantiating the SIMGRID platform and workloads in terms of jobs. Also, declaring and defining the services and controllers with all their logical conditions happens at this level. Last but not least, it launches the simulation and analyses it outcomes. In short, in this work, this last layer is the actual simulator.

The simulator in this thesis is based on the functions provided by WRENCH. However, since the HEP use-case demands complex platforms with many components and applications with special requirements, not all required functionalities are included in WRENCH. Others had to be bypassed in favour of improving the runtime and memory scaling of the simulator. In the following, for this thesis relevant components of WRENCH are presented.

Bare-Metal Compute Services – The bare-metal compute service is the most basic compute service implemented in WRENCH. Access to compute resources in order to execute workloads (usually in terms of jobs) is provided by this service. It is started directly on a host and gets the control over a configurable fraction of the CPU cores and RAM on that host. Optionally, storage space on that host can be configured as scratch space, where intermediate outputs of the applications executed by the service can be stored.

It simulates a daemon, which manages threads on the available cores and takes actions in case of thread failures. Also, it manages executors on these threads, when a workload is scheduled to the service, based on the requested number of cores. The overhead introduced by the starting of a thread and the status monitoring can be individually set or turned off on each service. In this thesis, the overhead is neglected and therefore turned off in all presented simulations.

Storage Services – The storage services in WRENCH simulate daemons that handle the access to storage resources on hosts. A storage service is started on a host and handles operations on the assigned storage resources. As such, it receives and answers file lookup and deletion requests and returns the status of these operations. For file write requests it checks for space on the resource and if successful handles the subsequent

data stream. Similarly, for file read requests it checks for presence of the file and handles the following data stream. All those requests and answers create load on the network between the client and the storage host and are therefore simulated. The load on the storage systems by the checks themselves create a compute overhead that is tiny and therefore neglected in the utilised model. For this thesis, however, the emphasis is not on the behaviour of real storage systems. Moreover, the messages in real systems are very small ($\mathcal{O}(10\,\mathrm{B})$) compared to the size of the processed data in HEP by each job. Therefore the size of those messages is kept default at 1 kB for all storage services. Also, since there can be more than one concurrent operation on a storage service, it is possible to restrict the number of concurrent data connections on the service. Per default, which is used in this thesis, there are no restrictions.

The data for read and write operations from and to the storage service is chopped into chunks of a configurable integer size between zero and infinity. This is done in order to adapt the read and write buffers of I/O applications in real systems. In the simulation, this is realized by a loop over a pipeline of storage I/O and network operations for each chunk of data with the size of the configured buffer. In the case of an infinitely large buffer size – or a buffer size equal to or bigger than the read or written file – the full I/O pipeline would simplify to a completely sequential process of a single storage read and network send or network receive and storage write.

It is possible to configure a zero buffer size. In this special case, a fluid model close to the original SimGrid model is implemented, which does not create a pipeline of individual chunked operations. Instead, the I/O creates a simultaneous load on both the storage resource and the network of the full data size which belong to the same activity, which proceeds at the bottleneck speed of all involved resources. Consequently, also here only a single operation needs to be simulated in exchange for a slightly more complex optimization problem in the assignment of resource capacity shares (see eqs. (6.1) and (6.2)).

HTCondor Compute Service – The HTCondor compute service in Wrench mimics the scheduling of and simulates the loads introduced by the HTCondor scheduler, which matches jobs to suitable compute resources (see section 6.1.2.2). Since HTCondor is a complex software consisting of many components with a lot of flexibility in the configuration, the implementation in Wrench is very simplified and only approximates certain parts.

The implementation in Wrench consists of a central HTCondor compute service, which is started on a host and gets a list of bare metal compute services assigned. These bare metal compute services correspond to the start services of the real system, while the HTCondor compute service can be used to submit jobs. Additionally, a central manager service is started, which manages the messaging between the starter and scheduler services and a negotiator service. The negotiator service is started by the central manager

and fed with information about the pending and running jobs and the available compute services which can accept jobs. Using this information it matches the pending jobs based on their requirements to fitting compute services with enough available CPU cores and memory.

Starting the services and sending the meta-data create load on the infrastructure. Therefore, WRENCH can simulate the exchange of messages and overheads. The size of these is configurable by the user. Comparing the introduced overhead by HTCondor in real systems ($\mathcal{O}(1\,\mathrm{s})$) to the typical run-times of the majority of HEP jobs ($\mathcal{O}(1\,\mathrm{h})$) the influence by HTCondor seems negligible. Therefore, for this thesis, the simulation of the overheads has been bypassed in order to restrict the number of simulated operations and consequently improve the speed of the simulation.

Execution Controllers – The execution controller in WRENCH is the base for any abstract process interacting with the WRENCH services. It already includes methods to create managers for data movement and job executions and creation. Furthermore, monitoring services, e.g. for measuring the instantaneous bandwidth or energy consumption, run and report within the execution controller. Concisely, every dynamic interaction with the platform and services while the simulation is progressing is modelled. It is achieved by hooking the execution controllers to the event chain of the simulation and defining reactions to be executed when a certain type of event occurs.

The freedom introduced by this variable concept of an execution controller is used to extend the simulation as described in section 6.3.2.1.

Jobs – The standard job class in WRENCH supports the simulation of jobs consisting of global input files to be processed, a chain of tasks with a certain amount of computational work per task and individual input and output files to be read and written. With this concept it is possible to create any workflow of read, compute and write operations, i.e. batch and streaming jobs as described in section 6.1.2.3 and fig. 6.1. However, this merely allows creating abstractions characterized by the amounts of data read, the computational work to be executed and amounts of data to be written. Dependencies between jobs, in order to be able building a workflow, are defined via the task dependencies. Extensions to this directed graph of tasks, e.g. additional logical dependencies or other types of operations, cannot be included. Also, the fact that in case of a streaming job, a high number of tasks have to be kept in memory, inflates the memory requirements during the simulation.

When more flexibility is needed, so-called compound jobs can be created. Those jobs are composed of individual actions which are connected by child-parent dependencies. Additionally, dependencies between jobs can be defined in order to build workflows. The actions available in WRENCH are file read, file write, file copy, file delete, compute, sleep and custom actions. The former model typical data operations on storage services, com-

pute operations on compute services and pause activity on a slot. The latter, the custom action, is a powerful abstraction, which allows the user of WRENCH to execute any desired procedure by the job during the simulation. This can be something very basic, e.g. changing or updating the content of a data container. But it can also be something complex, like executing real MPI code [215] on the simulated platform. This allows jobs to be an active component in the simulation, rather than just a passive entity. These features of custom actions are central to model the behaviour of HEP jobs in this work.

6.3.2 Simulator

For this work, SIMGRID and WRENCH have been chosen as the supporting frameworks for the simulation of HEP workloads on computing infrastructure designed for HEP, i.e. parts of the WLCG. The analytical models allow an efficient simulation, keeping the computational complexity reasonably low. At the same time, the models are complex enough to capture the key features of real LSDCS systems. However, several adjustments and extensions have to be implemented in order to be able to support the desired HEP features. Also, the actual simulator with a structure that enables the simulation of HEP applications needs to be defined. Lastly, the scalability of the simulator with respect to the size of the simulated platform and the number of simulated jobs has to be evaluated.

6.3.2.1 Extensions to WRENCH

In order to be able simulating and efficiently monitoring the simulated HEP applications a few extensions to WRENCH have to be implemented. The ones implemented for this study are presented in the following.

Streaming Jobs – The atomic unit of HEP workloads is a job. In general, these jobs read data, perform some computations based on this data and write output data. Both input and output data is typically not only locally read/written. Instead, data is transferred over the WAN of the WLCG. Conceptually, the jobs can be implemented using the default job actions provided by WRENCH (see section 6.3.1.2).

For batch jobs, which first copy the whole input data to local, process it afterwards and finally write some output data the according actions implemented in WRENCH can be used. Correspondingly, in order to be able defining such a job, the input data size V_{in} and location, the amount of computational work W_{comp} and the size V_{out} and location of the output data has to be specified. With this information, a batch job is fully characterized. However, batch jobs make up only a fraction of the workloads run in HEP.

As discussed in section 6.1.2.3, an integral part of HEP jobs stream input data to the executing core. Therefore, the execution of the job consists of a pipeline of multiple read, compute and write blocks. For a compute-block to start, the successful read-step of the corresponding input-data needs to be finished. The output-data of all blocks is typically gathered on scratch space before the total output data is transferred to the

desired destination. Consequently, there is a one-to-one correspondence between single respective read- and compute-blocks. Therefore, to characterize a streaming job the size of the blocks of input-data b_{xrd} has to be defined. In general, it is much smaller than the total amount of input data. Typically, the streaming block-size for HEP jobs utilising XRootD is at the order of 1 MB and the input data at the order of 1 GB. Hence, the number of blocks is given as

$$N_{\text{blocks}} = \lceil \frac{V_{\text{in}}}{b_{\text{xrd}}} \rceil \tag{6.4}$$

with the ceiling function denoted as $\lceil\ \rceil$.

It is assumed that the corresponding compute-block is directly linearly dependent on the size of the input-data block. This might not exactly correspond to real HEP jobs, since the input-data consists of independent events and the executing computations depend on the characteristics of each event individually. As a consequence, blocks randomly differ from each other in the amount of computational work to be executed since each block contains different events. However, when the block size is large and therefore each contains a big number of events, the computational amount of work per block approaches the arithmetic mean of computational work for the job per number of blocks. Therefore, for large block sizes and consequently large numbers of events in a block, the amount of computational work to be executed per block b can be approximated well as

$$w^b_{comp} = \frac{W_{\text{comp}}}{N_{\text{blocks}}} \tag{6.5}$$

with the total amount of work for the job W_{comp} and the number of blocks N_{blocks} given in eq. (6.4).

A direct effect of the choice for the block-size, originates in the pipelining nature of the streaming jobs. Similar to the buffer-size in the WRENCH storage services (see section 6.3.1.2), a streaming job creates a pipeline of individual chunks of operations on the network and a CPU core. The corresponding interaction in the storage service example would be a buffered write operation, where data comes from the network and has to be processed. For large block sizes, it results in a sequential process of read, compute and write, i.e. a batch job. Obviously, this is not desired. For a finite block size approaching zero, the amount of operations to simulate for each job increases. For zero block size, some kind of fluid model is conceivable. In contrast to the storage buffers, the pipeline in the job execution contains work accomplished by the network and CPU resources, which have different dimensions. Therefore, it is not trivial to implement a fluid model combining these two different quantities, which is left for future work. Consequently, a finite block size was chosen for this thesis, which needs to be small enough to enable a streaming behaviour.

The block size, however, cannot be too small. For small block-sizes, where the distinct composition of the events in the block matter, and jobs where the computation dominates the execution pipeline, the difference in the amount of computational work

per block might have a significant impact due to changes in the execution patterns of the jobs interfering with each other. This needs a dedicated investigation, which is out of scope for this thesis. Therefore, for simplicity, it is assumed that a block size of 1 MB is large enough for containing a statistical ensemble of events and the effect of block-dependent job pipelines is neglected. This assumption needs to be especially validated for the first steps in the typical HEP data processing chain with event sizes of up to 1 MB (see section 6.3.5.1). However, the contribution of such jobs is insignificant in the studies presented in this thesis.

Instead of creating the pipeline of read, compute and write actions of a streaming job using the classic WRENCH actions, a custom action which executes the chain of read and compute actions internally is implemented. In this way, only one action object plus the write action have to be kept in memory instead of $2N_{\text{blocks}} + 1$ actions per job.

Caching Action – Another benefit of a custom action is that any dynamic behaviour of jobs can be modelled. This is necessary for the simulation of data access and caching as implemented in XRootD (see sections 6.1.2.3 and 6.1.2.5) as part of the job, since both access and caching of data contain an active component in their initialization.

Input data locations are deduced by the XRootD cluster once a request for data is made. This needs a replication of the logic of the XRootD cluster of redirectors and servers in simulation, which would be an extension to the WRENCH storage service. Indeed, there exists a dedicated implementation as part of WRENCH which allows querying some XRootD storage service without a specification of the data location [216]. Subsequently, the XRootD storage service simulates the identification and redirection to the data server holding the data and the transfer from this server to the requesting job is started. This XRootD simulation in WRENCH originated from a simplified view of the XRootD logic, which has been implemented for this thesis. In this simplified implementation, the job queries each data server at the start and checks for the presence of the required data. Once a server is found, the data location is stored in the specification of the job. The lookup of the data is simulated without the creation of additional load on the network and storage services, but can be turned on by configuration if required.

Data caching needs to be triggered by a client requesting data from a server via a proxy. For the identification of the proxy, the job must be scheduled to an executing host, whenever there is a locality requirement on the definition of a cache. In the implementation of the simulator, three configurable locality scopes are implemented. Depending on the scope, only cache storage services on the same host, the same network zone or network zones in the same enclosing network zone are considered as suitable caches. In this thesis, only the local scope with the cache on the same host has been used. Once a suitable cache has been matched, the presence of the input data on the cache is checked. If it is present, the data is queried from the cache. If not, the data transfer from the storage service holding the data to the executing machine is duplicated

on the cache host and the next transfer to this data will be served from there.

When there is not enough space to cache incoming data, data present on the cache needs to be evicted. There are multiple decision algorithms of which data to evict, however, the standard is LRU. Therefore, a file list ordered according to the date of the last access of each file for each storage service is implemented. It is updated on each access to a file on a specific storage service without the simulation of any overhead. With this information, all caches can easily identify the least recently used files. These are evicted until enough space has been freed for caching the new incoming data.

In principle, the active cache behaviour could also be simulated as part of the XRootD storage service taking care of the data serving. However, the request of the data and the identification of the reachable cache need to stay job-dependent. This is left for future work.

Workload Execution – Instead of explicitly defining single jobs, jobs are often submitted in batches. In HEP jobs of the same batch execute the same code and differ only in the explicit input data they process. Indeed, most of the time this input data belong to the same dataset consisting of similar contents and differ only due to the probabilistic nature of HEP-collision events. As a consequence, the resulting characteristics of jobs of the same batch are very similar. Therefore, for simplicity the batches of jobs, called workloads, are configured.

A workload is characterized by the number of jobs it contains, the number of input-files its jobs process, the type of jobs (streaming or batch, see above), the submission time after the simulation start, the number of cores and amount of memory its jobs require, the amount of computational work to be executed and the size of the input- and output-files they read and write. To account for the internal variations between jobs of the same batch, the file sizes, the computational work and the core and memory requirements of the jobs can be modelled as arbitrary discrete or Gaussian probability density functions. When the job entities in the simulation are created, the explicit parameters are sampled from these probability distributions using MC methods (see section 2.2.1). However, it is also possible to create a workload with only one job and explicitly set its parameters to scalar values if desired.

In the simulator, for each workload a workload execution controller is spawned on a host. Each execution controller also gets an HTCondor service assigned to it. The execution controller starts becoming active when the submission time has been reached. Afterwards, the job entities are created from the randomly sampled or configured job specifications with the contained actions. Next, the jobs are submitted to the assigned HTCondor service, which takes care of the scheduling. Finally, once all jobs have been submitted, the controller hooks into the event chain and waits for all jobs to terminate.

In case a special type of event is thrown during the simulation, the controller takes actions defined by an event-type specific method. Since for this thesis it is assumed that all jobs terminate successfully, the simulation is aborted by the execution controller, when a job fails. When a job successfully completes, the execution controller clears metadata kept during the job creation and submission in order to decrease the memory footprint of the simulation.

Monitoring – Benefitting from simulation is only possible when the simulated traces are available for analysis. In WRENCH it is enabled by a built-in analysis framework. However, this framework does not support the monitoring of all types of interactions on the system. At the same time it requires significant amounts of memory and CPU, which scales badly with simulations of many jobs. Therefore, a dedicated monitoring was integrated into the simulator, which focuses on the monitoring of only the information that is relevant for this thesis in a crude but fast and resource-efficient way.

At the start of the simulation a file is created for storing the relevant monitoring information in the format of a table. In particular, the table's columns contain the job identifier, the name of the executing machine, the job start and end times, the input-file hit-rate on the reachable data cache, the time spent being processed by the CPU and the time it took transferring its input and output files. Since all those quantities can only be determined once the job has started or during its execution, they need to be cached first until fully determined and stored after completion of the job. The job start and end time can be easily determined at the job start and end event, this is more complicated for the other quantities, since they depend very much on the execution pattern of the specific job. Therefore, the determination of these is implemented as part of the same custom action handling the execution and caching. There, the necessary information for determining for instance the fraction of input files read from the cache, called the file hitrate, can be looked up with minimal effort on the matched cache storage service, once the data transfers start, following the definition of the hitrate of a job

$$h = \frac{\sum_{f \in \text{cache}} V_f}{\sum_{\forall f} V_f} \tag{6.6}$$

with input files f and a size in bytes denoted as V_f. For the calculation of the time spent in computing and data transfers, the start and end times of each execution and transfer block are determined, and the difference is incremented to the total computation and transfer times. The final values are then kept as metadata until the information is retrieved. When the job terminates, the cached metadata of the jobs is accessed by the responsible workload execution controller and the monitoring information for the respective job is written to the monitoring file. Finally, the metadata about the job is cleared.

6.3.2.2 Simulator Composition

The typical notion of a HEP workflow consists of jobs sent by users and collaborations of users to be executed on a computing infrastructure with the goal to produce physics results. A visualization of this is also depicted in fig. 6.7. Although there are some specific features in a HEP workflow which seem different from other communities' most of its attributes can be abstracted to a general picture valid for any type of workflow. Indeed, the differences only emerge in the value ranges of the characterizing parameters and specific realizations of the workflow dependencies and computing architectures. What all scientific computing workflows have in common, is that data needs to be processed. The processing is triggered by users of the computing infrastructure and they organize it by distributing the processing into individual jobs, which can be dependent on each other, building a workflow. The jobs are scheduled and executed on a computing infrastructure, where the data is transferred from data storage to the location where the computations are processed. Finally, when all the computations are finished a scientific output is harvested. The differences between communities are the user activity, the size of the data, the complexity of the computations and workflow dependencies, the scheduling logic and the computing architectures used. However, those differentiating features can be parametrized in a configurable way.

Although the simulator is designed to simulate primarily the execution of HEP workflows on arbitrary computing architectures, it is abstracted in a way to maximise flexibility in the configuration with the goal to enable the simulation of any combination of workloads and computing architecture as hypothetical scenarios. This is enabled by the models and tools provided by SimGrid (see section 6.3.1.1), Wrench (see section 6.3.1.2) and the extensions presented in section 6.3.2.1, which implement the general functions which build up the simulator, e.g. the dynamics of data storages and caches, the characterization, submission, and execution of jobs and the declaration and simulation of computing hardware architectures. The structure of the resulting simulator is presented hereafter.

The inputs which need to be provided to the simulator are the platform definition following the SimGrid standard (see section 6.3.1.1) extended with role assignments to each host and the configuration of the jobs and input data (see section 6.3.2.1). In preparation of the simulation, first, the simulator takes the configuration of the platform and creates a corresponding SimGrid hardware model. Next, the Wrench services get started on the hosts. The types of services started on each host is determined from the assigned role of that host. Worker hosts run bare metal compute services, storage, and cache hosts run storage services, scheduler hosts run HTCondor services, and executor hosts run execution controllers. Afterwards, the bare metal compute services get assigned to the HTCondor services. Also, the workload execution controllers (see section 6.3.2.1) get initiated with the assigned HTCondor services. Finally, the input files for all the jobs, which have to be present at the start of the simulation, get staged on the storages and a configurable fraction of those is preloaded on the caches.

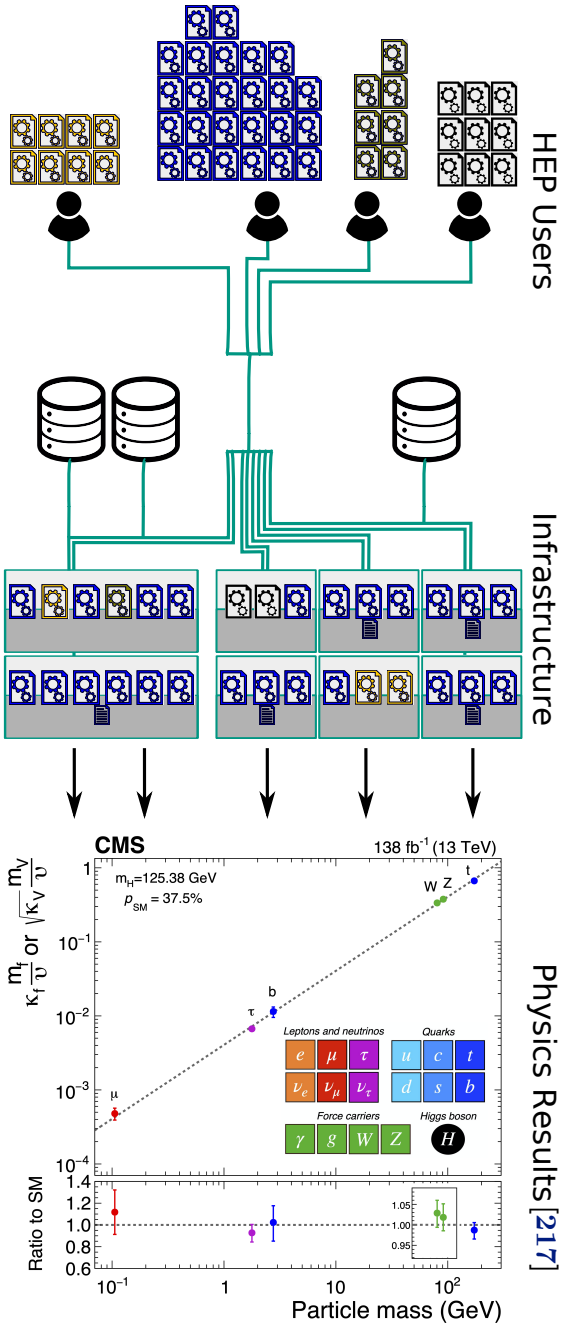


Figure 6.7: Schematic of a collection of HEP workloads executing on a computing infrastructure with the goal of producing physics results. HEP users, who can be individual users or collaborations, schedule all kinds of jobs on a computing infrastructure. The jobs are scheduled on the individual machines and are executed there. In the context of this thesis, the interaction of these components enables the successful processing of data measured by the LHC collaborations, e.g. CMS, on the computing infrastructure. Eventually, the processed results lead to the goal of inferring nature's principles, e.g. measuring the couplings of SM particles to the Higgs field [217] by CMS. The graphic was originally published in [208].

Once the above setup is done, the simulation is started. This means, the execution controllers become active and start creating and submitting the jobs. The jobs get scheduled by the according HTCondor services and start running on the bare metal services. Input data is transferred to the jobs and caches cache and evict data when applicable. When the jobs finish the output data is transferred to their destinations and the jobs terminate, making room for new jobs to be scheduled on the released slots. While this is simulated, the job traces are monitored and the significant information gets stored. This continues until all the jobs have been successfully simulated and the respective workload execution controllers terminate, leaving the user with a table of monitored job dynamics for analysis.

6.3.3 Calibration and Validation

There exists no established fundamental theory based on a set of first principles able to accurately describe the information flow in LSDCS with full complexity. Instead, all models (see section 6.2.2.2) are purely phenomenological, based on the notion of mimicking empirical observations. As such, they implement enough freedom in their conception, in terms of free parameters, in order to be able to adjust to the observations. Consequently, to provide a reasonable description of measured observables, the free parameters in the models have to be optimized or calibrated, similar to the tunes in HEP MC event generators (see section 2.2.4). This process is also known as calibration.

Once a simulator is calibrated, it is only validated to accurately predict the observables used in the calibration. Due to the freedom in the parameters a bias cannot be ruled out: The good description of the calibration data might be obtained by chance with a completely wrong model. To test the generality of the simulator, data independent of the data used in the calibration has to match with predictions obtained from a simulation. When it does not, either the calibrated set of parameters is in an odd configuration or the underlying models are wrong. When good agreement between the predictions and the validation data is observed confidence in the validity of the simulator is increased. This validation procedure is crucial to test the robustness of the models and their calibration. However, the validation of the models needs to be constantly revisited when new independent data is measured.

6.3.3.1 General Strategy

The strategy pursued in this thesis involves building several test architectures, with precisely known platform characteristics. On this computing platform workloads, consisting of jobs, with also precisely known characteristics are executed. In so doing, a simulation closely resembling the real system can be configured. This means that the platform architecture as well as the job characteristics in simulation are defined close to the parameters of the real-world system. This allows to start the calibration procedure from an initialized simulator which is expected to behave closely to the real system. However, this expectation needs to be continuously evaluated while performing the calibration.

A set of observables is defined, which is sensitive to the simulation parameters that need to be calibrated. In this thesis, the distribution of the jobs' execution times in multiple scenarios is studied. In a single measurement, the jobs' execution times are grouped according to the machine they were executed on and the median and the 2.5- to 97.5-percentile interval of each collection is computed. For each scenario multiple measurements are conducted to study the effects generated by the machine characteristics and influences on the measurement introduced by the remote system, which cannot be controlled. Without additional knowledge about these effects they cannot be modelled in a deterministic way and are therefore treated as random effects. Outliers, which show significant deviations in the distributions of the execution times are vetoed and not considered in this study.

The measured observables are then compared to the ones predicted by a simulation. When a significant deviation is found, the simulation parameters are adjusted and the procedure is repeated until the simulation matches the measured data. This can be done for several independent scenarios with different data taking conditions separately, each with enhanced sensitivity to a different set of parameters in the simulation. Consequently, a smaller amount of parameters of the simulation have to be tuned simultaneously, which simplifies the optimization procedure. However, the approach's success is strongly dependent on the ability to choose the right observables and scenarios and can only be performed with extensive knowledge about the simulator, its underlying models and the data used in the calibration. A direct approach, to optimize the full set of parameters in a single procedure combining all the measurements at once is more challenging. This would require a dense sampling of the parameter space with high granularity for finding an optimal description of the data. Since the amount of needed simulation runs scales exponentially with the number of parameters this approach scales poorly. Alternatively, an approach widely used in the tuning of MC event generators in HEP is to parametrize the behaviour of the simulator [39]. This helps with the scalability, since fewer points in parameter space have to be sampled, but in exchange for a bias introduced by the choice of the parametrization function used. It is left for future work to find a more robust but still accessible calibration strategy.

Since the simulation provides full control over its underlying models and is fully deterministic there is no need for repeating the same simulation. Instead, effects observed in the data, which are modelled by the calibrated simulation suggest the need for necessary extensions of the used simulation models.

Finally, a new set of data containing orthogonal information to the previous calibration, either due to be measured independently, originating from a new scenario or containing independent observables not used for calibration is used to validate the calibrated simulation.

6.3.3.2 Measuring Data

For calibration and validation the access to independently observed and precise data is crucial. This awareness is widespread in the scientific HEP community. As a consequence, theoretical and experimental physicists collaborate to build up and maintain an infrastructure for easily accessible data and metadata from many experiments for many years. This lead to a pool of available and further growing data, which can be used to tune, improve and check theoretical models. In the computing community, however, no such consensus exists.

The lack of available independent data poses a challenge for the research of complex LSDCS and stifles the advancement of theoretical models. Although there exist monitoring systems typically operated by the sites for accounting and quality assessment, which measure the state of their machines and the execution of workflows, the gathered information typically lacks network observables, since they are hard to measure, or is incomplete in other terms. Additionally, access to the data is restricted. In practice, this leads to limited usability. As a result, researchers are forced to measure their own data, which is typically tailored to their specific research problem. Although this can be sufficient for the calibration and validation of their models, testing their universality suffers from the limitations in the (scope of the) data.

Measurement Computing Architecture – For the measurement of calibration and validation data for this thesis four related test architectures are assembled, representing four independent scenarios. Although they differ in some specific characteristics described below, all four are based on the same general computing architecture:

Three machines of similar construction are connected via a switch to a local network. They are configured as worker nodes, able to accept and run jobs scheduled from a service machine in the same network configured as an HTCondor scheduler (see section 6.1.2.2). An HDD on each of the worker nodes with an XRootD proxy service is operated as a data cache (see section 6.1.2.5). A fifth machine used as a gateway is connected to the switch. All network transfers from and to the worker nodes are routed through this gateway. This enables artificially limiting the network speed for transfers to (from) an outside storage server from (to) the worker nodes via the network interface on the gateway. This is necessary, since there is no control over the shared remote link connecting the local zone with the storage server via the switch. In particular, the influence on the bandwidth by outside activities over the shared link cannot be reliably estimated. To circumvent this issue, by routing all communications through the gateway and restricting the bandwidth of the gateway's network interface to a value much smaller than the remote link's bandwidth, the effective remote link bandwidth for the local zone can be controlled. The exact characteristics of the utilised hardware is summarized in table 6.1. A sketch of the utilised architecture is shown in fig. 6.8.

The four scenarios are distinguished by steering two parameters of the system. First,

Table 6.1: Summary of the relevant hardware characteristics as advertised by the vendor or operator for the computing architecture used to measure the data for the calibration and validation of the simulator. The overhead introduced by the service machine and the switch is neglected.

Component	Relevant Characteristics
Worker Node 1	24 job slots, HDD 171 MB s^{-1} data transfer rate, 10 Gbit s^{-1} link to switch
Worker Node 2	12 job slots, HDD 171 MB s^{-1} data transfer rate, 10 Gbit s^{-1} link to switch
Worker Node 3	12 job slots, HDD 171 MB s^{-1} data transfer rate, 10 Gbit s^{-1} link to switch
Gateway	10 Gbit s^{-1} link to switch, 10 Gbit s^{-1} / 1 Gbit s^{-1} network interface
Remote Storage	80 Gbit s^{-1} I/O-bandwidth, 2 × 100 Gbit s^{-1} link to switch

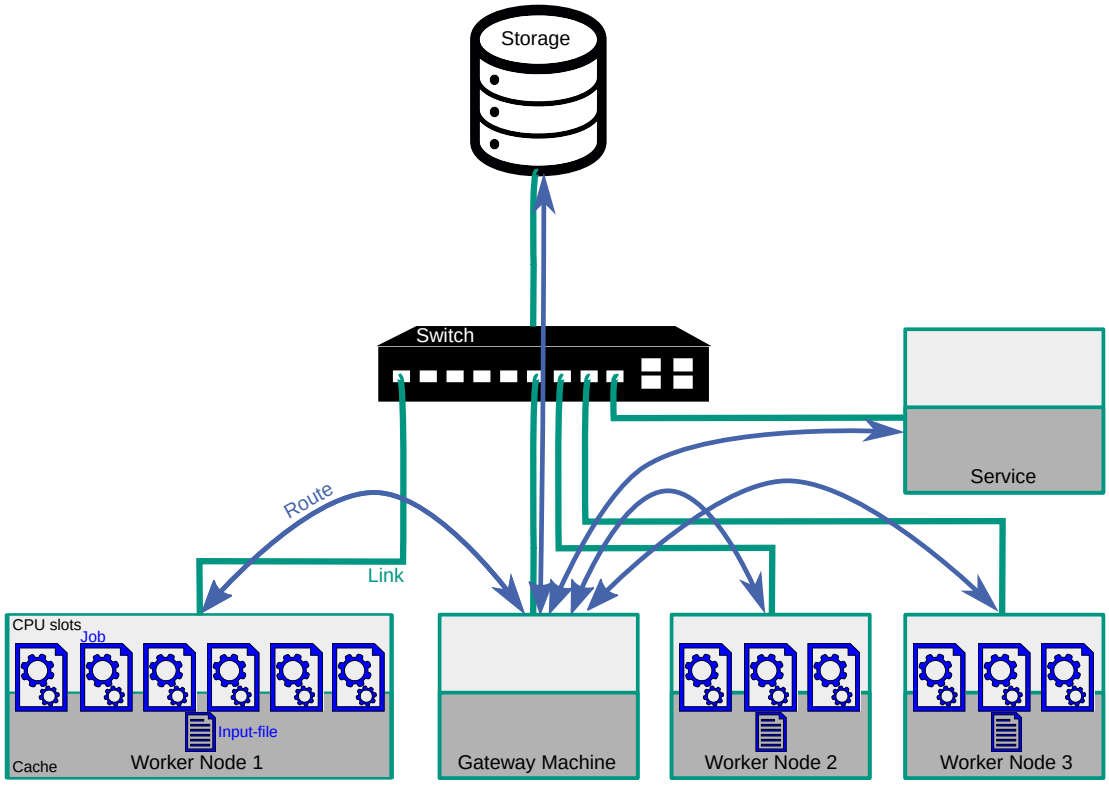


Figure 6.8: Sketch of the computing architecture used for measuring the data used in the calibration and validation of the simulator. Transfers between a remote storage server and three local worker nodes with internal data caches are routed through a local gateway machine. This enables steering the effective bandwidth of the remote link through the network interface of the gateway. A local service machine is used to schedule jobs to the worker nodes. The jobs read their input data from the cache or the remote storage depending on the scenario.

the speed of the network interface of the gateway machine can be adjusted to create a scenario with a fast and a slow connection of the worker nodes to the storage server. Second, the speed of the data cache on the worker nodes can be adjusted by enabling the page cache on the worker nodes. Thus, one can create a scenario where cached data is read from the slow HDD or a scenario where it is provided by the much faster RAM. The combination of the two options for these two parameters leads to a total of four different platform scenarios. Executing the same workloads on each of the four varied platforms respectively, enables measuring four sets of independent data with straightforward relation, which simplifies the calibration and validation of the simulator.

Measurement Workload – The workload executed on the test architecture consists of 48 jobs requiring a single core for execution each. This corresponds to the maximum number of job slots advertised by the HTCondor worker nodes assuring a full occupancy of the machines. Consequently, the potential load on the whole platform including the network is maximised, which helps to create scenarios in which specific parts of the platform limit the execution of the workload.

The application each job processes is based on an executable used in the preparation of data measured by the CMS collaboration for the analysis [218]. All jobs execute the same application and stream and process (replications of) the same data accessed over network or at the local disk via XRootD. In this way, the complexity is reduced since the precise knowledge about one job's characteristics is sufficient for characterizing the whole workload. In addition, each job announces the same requirements to the resource scheduler assuring the same load introduced by the scheduling of each individual job.

The only difference between individual jobs is introduced by the monitoring. Monitoring systems measuring relevant aspects of the execution of jobs from outside interfere with the system they have to monitor. Consequently, they introduce load on the system which is hard to model in detail. To evade this additional layer of complexity, observables relevant for this study, in particular the job execution times but also other quantities for cross-checks for example the read and written amount of data, are measured during the execution of the jobs as part of their executable. The measured information is then transferred as an output of the job. Since the execution pattern of each job depends on the dynamic state of the computing system it is processed on, the monitoring following the execution pattern differs between individual job executions. However, the monitored application is computationally more expensive by several orders of magnitude and transfers a factor of $> \mathcal{O}(10^6)$ more data than the monitoring. The loads introduced by differences in the monitoring between individual jobs is smaller or equal to the load introduced by the monitoring. Therefore, the loads introduced by the differences in the monitoring are neglected.

Summarizing, a single job is fully characterized by the number of operations to be executed, the amount of data to be read and written and its CPU and memory require-

Table 6.2: Overview of the workload configuration for the calibration and validation of the simulator. The values have been measured by running a test batch of the jobs at a machine with known and stable computational speed and access to the network interface providing information about the read and written bytes over network. The obtained values are rounded to a permille precision.

Quantity	Value
# Req. CPU cores	1
FLOP	2.886 T
Memory	2.4 GB
Input data	8.554 GB
Output data	16 MB

ments. The values have been measured by running a test batch of the jobs transferring the full data over network at a machine with known computational speed. Although the obtained values for the amount of operations performed for the specific executable were not exactly the same for repeated measurements, they coincided at a level of permille precision. Readout of the network interface provided information about the read and written bytes over network, which in the used application are the only relevant input and output activities. Here too, repeated evaluations coincided at a level of permille precision. The values obtained are shown in table 6.2.

Variations of this workload are constructed in order to create scenarios with different sensitivity on the simulation parameters. The scenarios differ in the fractions of data read via XRootD from network and local disk. The data read consists of 20 individual files. All files are accessible on the remote storage as well as on the local disk of each worker node. The job executable reads the input data via XRootD and takes the files to process as arguments. By setting the XRootD paths accordingly, the source location of the files is controlled. This is done for all the jobs in the scenario. The workload is repeatedly executed for eleven preset fractions of files between zero and one read from the local disk on the same platform.

6.3.3.3 Calibration

The four different scenarios for the platform together with the eleven workload scenarios results in a total number of 44 distinct scenarios potentially contributing to the calibration. Since the execution time of the streaming jobs (see section 6.1.2.3) chosen as an observable are a product of the speed of the execution and the data transfer, it provides sensitivity to both the configured speed of the executing machines and the effective available bandwidth for the jobs' transfers in simulation. However, with this information alone these two effects cannot be distinguished. By varying the amount of data read via network or from the local disk, sensitivity to the available network and disk bandwidth is added. Combining the information allows to unfold the influence of the computation and data transfer. Consequently, by combination of the eleven workload scenarios for

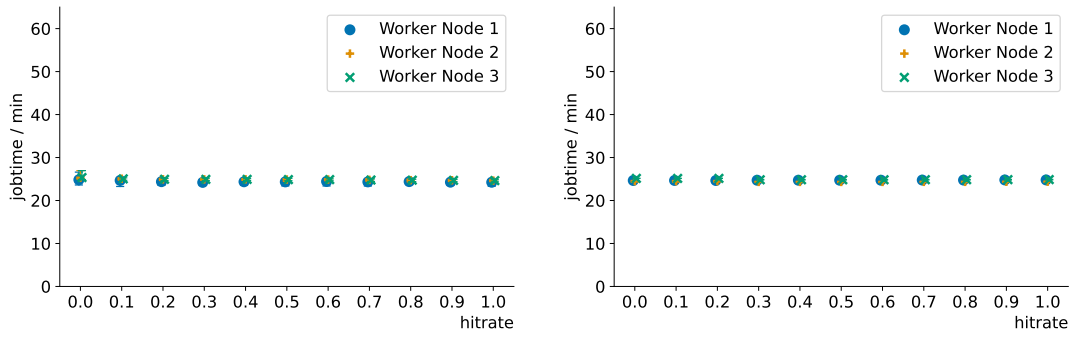
a single platform scenario the models used in simulation for the computation, the data transfers via network and from disk can be calibrated.

The four different platform scenarios impact the network and disk bandwidths for the data transfers. Therefore, a change in the quantitative contributions of the transfer and computation times to the jobs' execution times are expected. This can be exploited to refine the calibration in multiple distinct steps. Starting with the most sensitive scenario to a specific subset of parameters in the simulation models, this can be used for tuning of these parameters resulting in a preliminary calibration. In the next steps, the other scenarios less sensitive to a specific parameter subset can use the preliminary calibration(s) to fix some parameters increasing the sensitivity to the subset most relevant for the respective scenario. Consequently, a full set of calibration parameters is obtained with manageable complexity in each individual calibration step.

As presented in the following, three calibration steps to tune the simulation parameters sensitive to the computational speed, the effective bandwidth of the network and the effective bandwidths of the data caches are performed. They utilise the data measured in three of the four platform scenarios. The data obtained in the remaining platform scenario is saved for validation. Other simulation parameters, i.e. the size of the streaming blocks (see section 6.3.2.1) and the buffer size of the storage services (see section 6.3.1.2) are fixed to 100 MB and inf. Since the observed influence by these parameters on the simulation result is much smaller than the parameters presented in the following calibration, their superimposed effects on the observables are covered by the calibration procedure below. Further studies on the fine-tuning of these parameters is left for future work.

Step 1: Computation – First, the data measured in the scenario with the faster network interface bandwidth at the gateway of $10\,\mathrm{Gbit\,s^{-1}}$ and operative memory cache is used. Since this configuration maximises the available bandwidths for the jobs both on the network and on the local storage on the worker nodes operated as a data cache the influence of the data transfers in the job executions is minimized. This makes this scenario the best suited for calibrating the computation model in the simulator.

The medians and the 2.5- and 97.5-percentiles of the execution times of the jobs obtained from repeated measurements for this platform scenario are depicted in fig. 6.9a. Separate observables for each worker node are shown. The different workload scenarios with altered fractions of input-files read from the cache, denoted as hitrate in the following, are shown in different bins. It can be observed, that there is only little difference in the execution times of the jobs for all worker nodes and hitrate bins. The flat dependency of the observables on the hitrate indicates that the execution of the streaming jobs is completely confined by the speed of the executing CPUs. Since a change in the amount of input data read from two differently fast sources does not lead to a change in the execution time, it can be concluded that in this scenario the network



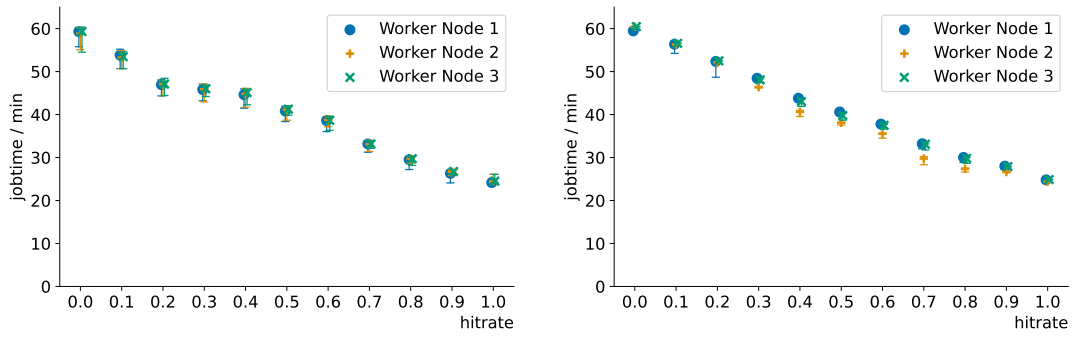
(a) Measured **(b)** Simulated

Figure 6.9: Comparison of the measured calibration observables (left) with the prediction obtained from the partially calibrated simulation (right) for the computing platform with a high gateway interface bandwidth of 10 Gbit s^{-1} and fast memory cache. In both plots, the median (points) and the 2.5- and 97.5-percentiles (whiskers) are shown for the execution times of the jobs (jobtime) separately for each executing machine in bins of the fraction of input-files read from the cache (hitrate).

and cache bandwidths are big enough to feed the job with data faster than the data can be processed on the CPU. Consequently, this data can be used only for tuning the computation model in the simulator, i.e. the speed of the CPUs. There is no handle on the parameters of the network and storage models, since only a lower limit on the speed of the effective bandwidth to the remote storage and the bandwidths of the local and remote storages can be set. Since there is only a little dependence of the execution times on the executing machine observed, the CPU speeds of the CPUs on the worker nodes are similar. That is to be expected, since all three worker nodes are built up from the same hardware components. However, zooming in tiny differences can be observed.

For the first calibration step the remote network bandwidth and the bandwidths for the storage services running on the worker nodes representing the caches in the simulation are set to large values $\gtrapprox$ 11.5 Gbit s^{-1} and $\gtrapprox$ 800 Mbit s^{-1}. Below that, the simulation starts to show an influence on the job execution times in high and low hitrate bins, which has not been observed in the measured data. Next, the speeds of the CPUs in the simulated platform are repeatedly adjusted by hand until the simulation reproduces the observed job execution times. The in this step obtained calibration is depicted in fig. 6.9b.

It can be observed in comparison of the measured and simulated observables after the first calibration in fig. 6.9 that the simulation of the first platform scenario agrees well with the measured data. For this the naive initial CPU speeds have to be decreased by $\approx 20\%$. It is unclear, whether this is necessary due to the poor estimation of the CPU speeds or the characterization of the jobs. However, both cases can be corrected by the constant calibration factor.



(a) Measured **(b)** Simulated

Figure 6.10: Comparison of the measured calibration observables (left) with the prediction obtained from the partially calibrated simulation (right) for the computing platform with a low gateway interface bandwidth of $1\,\mathrm{Gbit\,s^{-1}}$ and fast memory cache. In both plots, the median (points) and the 2.5- and 97.5-percentiles (whiskers) are shown for the execution times of the jobs (jobtime) separately for each executing machine in bins of the fraction of input-files read from the cache (hitrate).

Step 2: Network – Second, the data measured in the scenario with the slower network interface bandwidth at the gateway of $1\,\mathrm{Gbit\,s^{-1}}$ and still operative memory cache is used. Compared to the platform scenario used in the first calibration step an increasing influence on the execution times by the transfers over network due to the decreased bandwidth is expected. Since the bandwidth to the local data caches on the worker nodes is kept the same in this scenario, there should be no effect visible as already observed in the previous calibration step.

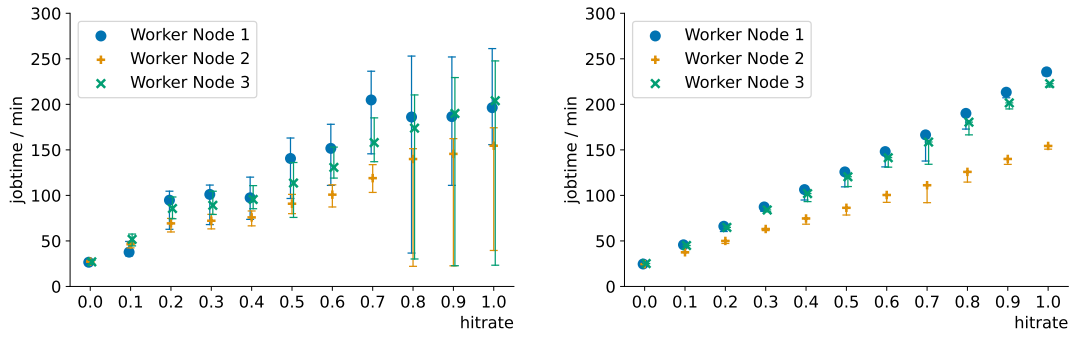
The medians and 2.5- and 97.5-percentiles for this platform scenario's measurements are shown in fig. 6.10a. A clear dependence of the execution times on the fraction of files read from the cache can be observed. The execution time is maximal when all the input data is read via the network from the remote storage and decreases with higher fractions of the hitrate, which consequently means less data is read via network. This indicates a strong dependency of the jobs' execution times on the available bandwidth. The total bandwidth shared among all running jobs is small enough that the CPUs processing the jobs have to wait for the input data blocks to be streamed. At a maximum hitrate, all input files are provided by the fast memory cache on the worker nodes. Indeed, the last bin for a hitrate of one with the measured execution times in fig. 6.9a shows perfect agreement. Therefore, a linear dependence on the amount of bytes read via the network is expected. This expectation is compatible with the observed dependency in fig. 6.10a. Since here, the hitrate is the fraction of files on cache and the file sizes are not equal, converting the hitrate to a hitrate of bytes would result in a shift of the measured bins for hitrates of 0.2 and 0.4 closer to a byte-hitrate value of 0.3, resulting in the expected linear dependency.

In this simulation, all input file sizes are equal. Therefore, the byte-hitrate is equal to the file-hitrate in simulation, which directly leads to a visible linear dependency of the execution times on the hitrate. By varying of the bandwidth to the remote storage in simulation the slope of this linear dependency can be adjusted. Based on the preliminary calibration obtained from the previous calibration the network bandwidth is repeatedly adjusted. The resulting hitrate dependencies are compared to the measured ones and the adjusted value that matches best is determined for the calibration in this step.

Consequently, to obtain the best fitting simulation results, which can be observed in fig. 6.10b, the naive network bandwidths in the initial simulation platform need to be increased by $\approx 15\%$.

Step 3: Data Caches – Third, the data measured in the platform scenario with the fast network interface bandwidth of $10\,\mathrm{Gbit\,s^{-1}}$ and enabled HDD cache on the worker nodes is utilised. The bandwidth of the HDD cache is expected to be much slower than the bandwidth to the memory cache. Consequently, for the same arguments as discussed in the previous step an effect introduced by the slow bandwidth of the HDD cache is expected for high hitrates, while the network bandwidth is high enough to present no visible effect across all hitrates.

The medians and 2.5- and 97.5-percentiles measured for this platform scenario are shown in fig. 6.11a. As expected, with increasing hitrate and therefore a higher fraction of files read from the cache the low bandwidth of the HDD caches increasingly throttles the execution of the jobs. Consequently, since during job execution the processing has to wait for the data to be transferred from the cache the job execution time increases with a higher fraction of data read from the cache. As in the previous step, the measured observables are shown in bins of the file-hitrate, not the byte-hitrate which would move the observed points around 0.3 closer together in hitrate. Another observation is that with increasing hitrate the spread of the job execution times indicated by the depicted percentiles increases significantly. This can be explained by the properties of the HDD storages. From the point of view of the HDD disks multiple jobs running on a worker node try to concurrently access different data on the same disk. This data is stored on different physical positions on the magnetic drive. For random access of the data on the physical drive, the moving actuator arm with the magnetic head responsible for data readout needs to switch the position. This random mechanical movement on the one hand limits the speed in random access compared to a sequential read procedure and on the other hand leads to high variance in the access speed of specific data. Therefore, with increasing number of data read from the cache more concurrent random accesses to the data on the HDDs occur, which increases the variance in the received byte rates for each individual data access. Consequently, the spread in the job execution times increases as well. When replacing the HDD with a flash storage this spread is not expected to be observable. Checking this hypothesis is left for future work.



(a) Measured **(b)** Simulated

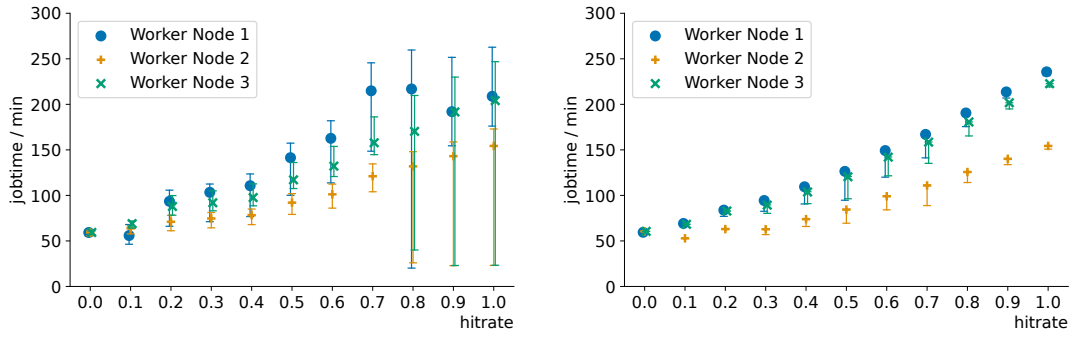
Figure 6.11: Comparison of the measured calibration observables (left) with the prediction obtained from the fully calibrated simulation (right) for the computing platform with a high gateway interface bandwidth of 10 Gbit s^{-1} and slow HDD cache. In both plots, the median (points) and the 2.5- and 97.5-percentiles (whiskers) are shown for the execution times of the jobs (jobtime) separately for each executing machine in bins of the fraction of input-files read from the cache (hitrate).

This internal behaviour of HDD is not considered in the data access on storage model implemented in the simulator (see sections 6.3.1.2 and 6.3.2.1). Therefore, the simulation cannot reproduce the increase in the spread of the job execution times. However, the simulation is able to predict the general dynamics with its implemented models. Therefore, the simulation is tuned to reproduce the median of the observed job execution times. For this, based on the calibration obtained in the two previous steps the bandwidths of the storages on the worker nodes are varied in simulation. The obtained observables are then repeatedly compared to the measured ones shown in fig. 6.11a and the bandwidth values applied in the best fitting variation are kept as the final calibration.

The obtained bandwidths in the best calibration shown in fig. 6.11b are $\mathcal{O}(10)$ lower than the naive initial values, which were estimated by the vendor of the HDDs. This is due to the concurrent read operation of the running jobs, which lead to the much lower data read rate. Additionally, although the utilised HDDs are structurally identical on all worker nodes, differences of up to $\approx 90\%$ in the calibrated bandwidths in simulation are obtained.

6.3.3.4 Validation

After the calibration the obtained observables from the fully calibrated simulation needs to be compared to a set of data independent of the data used in the calibration for validation of the simulation models. Therefore, the data measured in the remaining platform scenario is utilised. In this scenario, both the gateway to the remote network is throttled to 1 Gbit s^{-1} and the slow HDD caches are utilised. This presents the simulator with the challenge of predicting a scenario, in which both the bandwidths of the network and



(a) Measured **(b)** Simulated

Figure 6.12: Comparison of the measured validation observables (left) with the prediction obtained from the fully calibrated simulation (right) for the computing platform with a slow gateway interface bandwidth of $1\,\mathrm{Gbit\,s^{-1}}$ and slow HDD cache. In both plots, the median (points) and the 2.5- and 97.5-percentiles (whiskers) are shown for the execution times of the jobs (jobtime) separately for each executing machine in bins of the fraction of input-files read from the cache (hitrate). This has been presented for the first time in [208].

the caches are a limiting factor for the execution time of the streaming jobs. The models implemented in the calibrated simulation are only valid when this unseen data can be correctly reproduced in the simulation.

The medians and 2.5- and 97.5-percentiles measured for this validation platform scenario are depicted in fig. 6.12a. According to the expectations, it can be observed, by comparison of fig. 6.11a and fig. 6.12a, that the slow network bandwidths influence is the greatest for small hitrate values. For higher hitrates, although the bigger effect of the much slower cache bandwidths starts to dominate the execution time the slow network still leads to higher job execution times for hitrate values < 1.0. As a consequence of the slow HDDs and the concurrent data access, the observed increasing spread in the execution times with higher hitrate remains. As already discussed in section 6.3.3.3 this is not modelled in the simulator. Therefore, only the medians of the observables are expected to be predicted correctly by simulation.

A platform configuration that takes all the calibrations into account, shown in appendix B.2.1, is used to recreate the observables measured in real-world data. Without any further changes to the simulation parameters, the simulated observables shown in fig. 6.12b correctly reproduce the measured observables. Therefore, it can be concluded, that not only the calibration was successful, but also that the models implemented in the simulation are robust. This simulator calibration is therefore used in the studies presented in this thesis.

6.3.4 Computational Complexity of Simulation

Accurately simulating LSDCS is the necessary condition. This is achieved by modelling the systems with robust, calibrated and validated models. However, the challenge is to at the same time also fulfil the sufficient condition: Being able to complete a simulation under a time budget. As mentioned in section 6.2.2.2, models that integrate many aspects of complex systems tend to describe those with more accuracy than simplified surrogates in exchange for a more complex simulation and therefore longer simulation runtime. This trade-off between simulation accuracy and runtime has to be kept in mind when designing a simulator. Therefore, the scaling of memory and runtime of the presented simulator in section 6.3.2 with the number of simulated jobs and the size of the simulated platform is studied in the following. Since the exact execution time and memory usage depends strongly on the machine architecture the simulation is executed on as well as on the configuration of the simulator, the results obtained in the following studies are subject to potentially high variations, which have not been studied. However, the observed trends and qualities of the simulator should apply to any simulation obtained with this simulator.

6.3.4.1 Simulation Scaling of a Small Platform

To start the study of the scaling of the simulator the platform specified in appendix B.2.1 is used. The platform contains a local network zone with three hosts dedicated to run jobs with a total amount of 48 CPU cores connected to a shared network switch. Data is served by and written to a remote storage server connected via a single link to the switch. Several simulations with increasing numbers of jobs are started.

The jobs' characteristics follow the configuration given in table B.1 inspired by a benchmark workload used in [219] on a similar system. The amount of computational work and the sizes of the input and output files are drawn from Gaussian distributions, each with a standard deviation of 10% compared to the nominal value. The computational work is given in units of floating point operation (FLOP)s, the sizes of the files in Bytes. The jobs stream and compute their input data in blocks of 1 GB, which is approximately 100 times larger than the typical size of a block in an application reading inputs via XRootD. This choice reduces the overall runtime of the simulation, since fewer operations have to be simulated compared to a more realistic simulation with a smaller block-size.

Since every simulated job has to be individually spawned and kept in memory during the simulation and each job object is of a similar type an approximately constant amount of memory per job is expected to be occupied by the simulation. This leads to a linear growth in the maximum of utilised memory with the number of simulated jobs. Additionally, an offset in utilised memory due to the initialization of the other simulation objects and the instantiation of the platform which are active during the whole execution time of the simulation is expected.

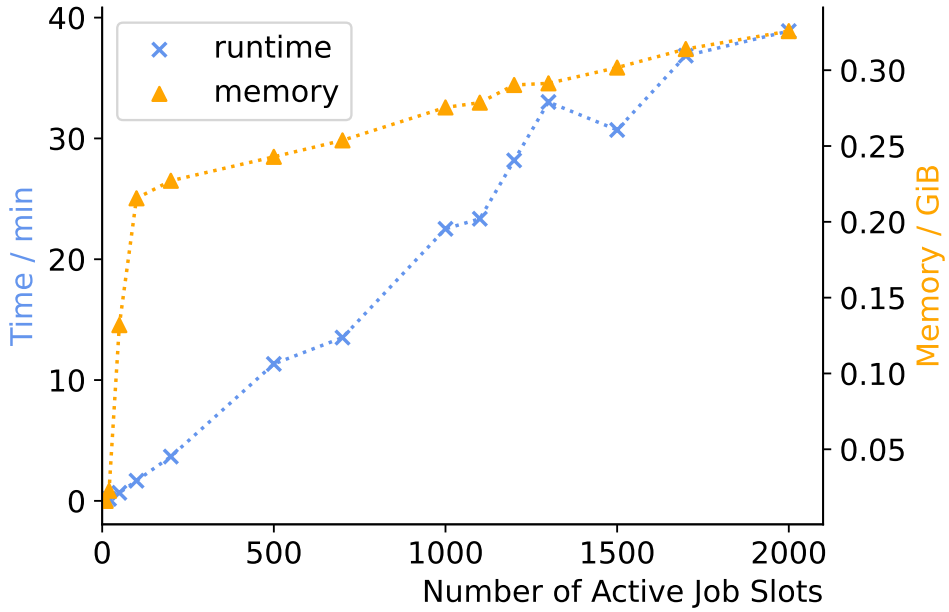


Figure 6.13: Memory maximum and runtime scaling of a simulation of an increasing number of simulated jobs running on a platform with $\mathcal{O}(10)$ cores. As expected, the increase with the number of simulated jobs of the execution time as well as the utilised memory is compatible with linear growth.

Above the threshold of 48 simulated jobs, which for this type of workload corresponds to the maximum number of concurrently running jobs in simulation, a linear growth is expected as well. Since the individual jobs are similar in their characteristics, during the runtime of the simulation approximately the same number of jobs and therefore the same number of operations have to be simulated. Deviations are only expected at the end of the simulation, when the job queue is emptied and fewer jobs remain running without new ones following. As a consequence, fewer operations would have to be simulated with smaller numbers of concurrent activities, leading to a simpler determination of the max-min-objective in SimGrid (see eqs. (6.1) and (6.2)), towards the end of the simulation. However, since the individual jobs are very similar, the size of the time frame in which this occurs is expected to be small. Also, since the jobs are drawn from the same distribution the effect on the execution time is expected to be of the same size for all simulations where the number of simulated jobs is a multiple integer of the threshold. Therefore, a linear growth in the execution time of the simulation with the number of simulated jobs is expected.

The measured execution times and the maxima of memory utilised by the simulations over the number of simulated jobs is depicted in fig. 6.13. According to the expectations, a linear growth with the number of simulated jobs in both the utilised memory

and execution time is observed. An extrapolation of the observed scaling to a number of $\mathcal{O}(10^6)$ simulated jobs would lead to memory requirements of $\mathcal{O}(100\,\mathrm{GB})$ and a runtime of $\mathcal{O}(100\,\mathrm{h})$.

6.3.4.2 Simulation Scaling of Large Platforms

With an increase of the size of the simulated platform and its utilisation, more simultaneously active activities need to be considered during the simulation. As a consequence, the max-min-objective in SimGrid is expected to become computationally harder to solve. Therefore, below the threshold of full occupancy of all CPUs an increase in the simulation runtime is expected with increasing numbers of jobs creating activities on the simulated platform. After the full occupancy is reached, the simulation is expected to scale as described in section 6.3.4.1.

For this study, the simulated platform has been changed to a similar configuration as in appendix B.2.3. But all numbers of CPU cores and link and storage bandwidths have been scaled with a factor 100. This results in a platform containing clusters of machines with 62,000 CPU cores able to run jobs, which is three orders of magnitude larger than the platform studied in the previous section.

In order to study the scaling of simulation execution time and maximal memory utilisation, the previous workload is utilised again. Again, several simulations with increasing numbers of jobs are started. However, due to the size of the platform fewer jobs are started than can concurrently run in the simulation and the scaling of the simulation towards the threshold of full occupation is investigated.

Following the discussion in the previous section, the increase of the maximally utilised memory by the simulation with the number of simulated jobs is expected to be linear. The scaling of the execution time below the threshold of full occupation of the CPU cores in simulation, however, is expected to be influenced by the increasing amount of concurrent activities, which need to be handled during simulation. The amount of concurrent activities is proportional to the number of concurrently active jobs in simulation, creating loads on the simulated platform. With increasing amounts of concurrent activities solving the max-min objective in SimGrid (see eqs. (6.1) and (6.2)) becomes computationally more laborious. Increasing the number of simulated jobs above the threshold of full occupation, the number of concurrent activities creating load on the platform does not increase, since the total number of running jobs at any point in time in simulation is capped and only these jobs create activities during their execution. In simulation the activities on the platform created by the queued jobs can be neglected compared to the activities introduced by the executing jobs.

The measured execution times and maximally utilised memory for the started simulations are shown in fig. 6.14. Since each running job occupies a fixed part of the executing machine based on its requirements, below the threshold of full occupation the number of

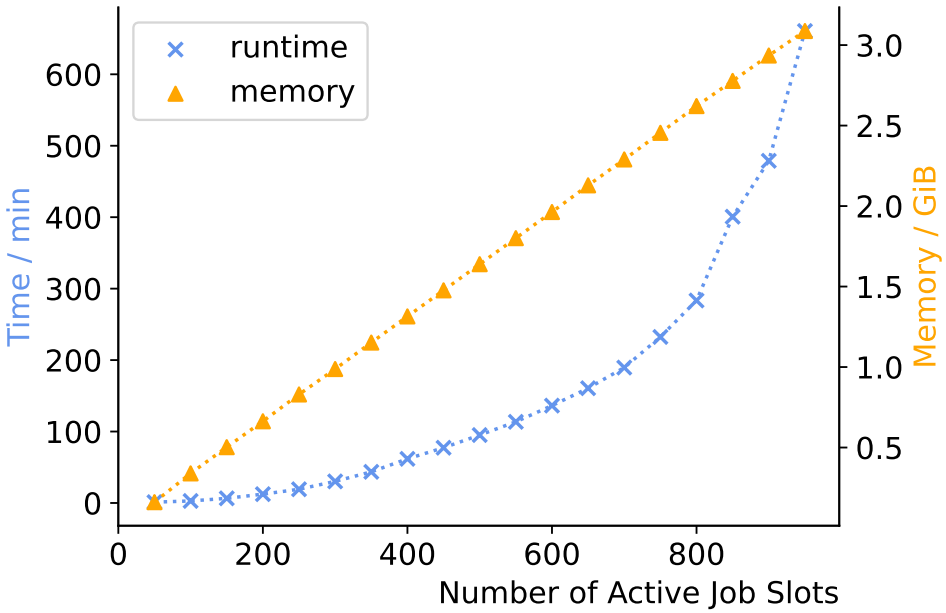


Figure 6.14: Memory maximum and runtime scaling of a simulation of an increasing number of simulated jobs running on a platform with $\mathcal{O}(10^5)$ cores. Below the threshold of full occupation each running job occupies a slot on the executing machine creating activities. This is defined as an active job slot. As expected, the increase with the number of active job slots of the maximally utilised memory is compatible with linear growth. The scaling of the execution time is superlinear, due to the increasingly costly solution of the max-min objective in SIMGRID presented in eqs. (6.1) and (6.2). This plot was originally published in [208].

started jobs in the simulation corresponds to the number of such active job slots in the simulation. These active job slots are the driver of the observed scaling. As expected, the memory utilisation increases linearly with the number of active job slots. For the scaling of the execution time with the number of active job slots a superlinear increase is observed. This matches the expectations of an increasingly expensive computation of the max-min objective in SIMGRID due to a higher number of concurrently contributing activities.

Extrapolating the observed execution time of the simulation to a fully occupied platform of $\mathcal{O}(10^5)$ active job slots leads to values beyond feasibility independent of the superlinear extrapolation model. Although above the threshold a linear increase in simulation execution time with the number of simulated jobs can be expected, the slope of the scaling would be dictated by the time a fully occupied simulation needs for execution. Since this increases superlinear with the number of active job slots and is therefore proportional to the size of the simulated platform, the simulation of large platforms, for instance LSDCSs like the full WLCG, becomes a computationally expensive endeavour.

To still be able to simulate at least sections of the WLCG in reasonable time workarounds have to be found.

6.3.4.3 Example Scaling Solution

One possible workaround, which retains the accuracy of the SIMGRID model with a significant boost in execution time and is used in this thesis for the study of a hypothetical WLCG scenario in section 6.3.5.2 exploits the superlinear scaling of the simulation execution time in reverse. Since the computational complexity increases superlinearly with the number of concurrently active job slots and only linearly with the number of simulated jobs most can be gained by limiting the former in a large simulated platform.

The underlying hypothesis is: The significant effects on the job execution in a LSDCS come from the internal structure of the network defining the complex equations of motion responsible for the dynamics of the system. As long as the relative scales of the individual components to each other are conserved, the dynamics does not change. Indeed, the dynamics are invariant under global scaling of the platform parameters. For a large platform with many potential job slots, but a comparably simple structure, the number of job slots can be globally scaled down by for example decreasing the number of cores and RAM on the hosts. Additionally, the bandwidths for the network links and storages need to be scaled down accordingly, to conserve the bandwidth per core. Finally, the number of simulated jobs needs to be scaled down accordingly. This presupposes that the collection of simulated jobs can be interpreted as a (group of) statistical ensemble(s) individual jobs can be sampled from. So the general characteristics of the simulated jobs as a whole does not change except for statistical effects by reducing the number. This significantly limits the number of concurrently active job slots in the simulation, which consequently reduces the runtime of the simulation.

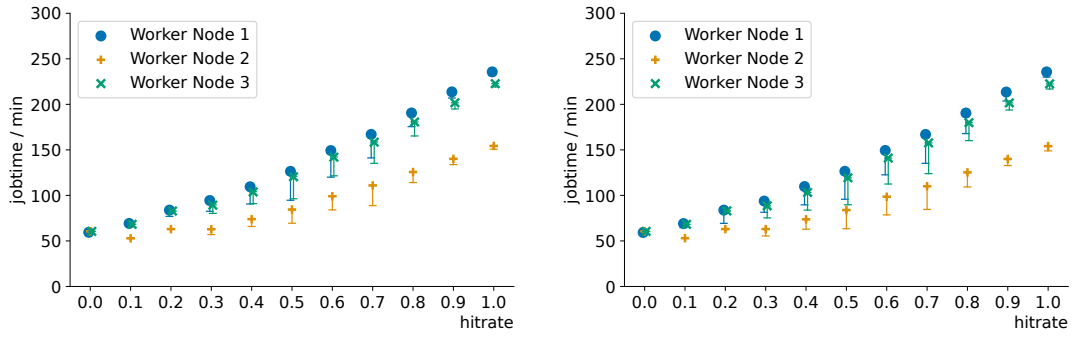
The limitations of this approach are twofold. First, if the collection of jobs is either not big enough or there are too many relevant clusters in terms of their characterization each consisting of a low number of jobs, the interpretation of the workload as (multiple) statistical ensembles breaks down. In that case, this approach is not applicable at all. Second, the scaling of the platform can only be driven so far until only single integer job slots remain on the hosts. Further reducing the size of the simulated platform would change the dynamics of the system, since a jobs requirements previously matching might not be met after the scaling any more. The scaled worker node would not accept that job type any more, and they would need to be executed elsewhere. This might lead to differences in the dynamics of the whole system. Summarizing multiple worker nodes in vicinity in terms of the network, for example combining multiple hosts of the same cluster into a bigger host, can extend the scaling potential. However, when the structure of the network is relevant for the dynamics of the workloads on that platform the simulation on the platform with merged host leads to different results. Therefore, the elimination of internal structures for mitigation of the computational complexity of the simulator needs to be carefully examined on a case-by-case basis.

Proof of Concept – In order to test this approach, the validation data presented in section 6.3.3.4 is revisited. As has been shown, the simulator is able to produce valid results with a dedicated validation platform used in simulation. When the hypothesis of the scaling approach holds, scaling the platform while retaining the complexity of its structure and repeating the simulations would result in the same predictions as presented in fig. 6.12b.

Since the size of the validation platform is already small and further decreasing it in simulation would remove interferences between the jobs executed on the same host it cannot be reduced further without running into the lower platform complexity limit discussed above. Therefore, to study the scaling the platform is scaled up instead of down. Since the hypothesis works in both directions and there is no limitation to the up-scaling, this does not change the validity of the study.

The validation study presented in section 6.3.3.4 is repeated with the scaled platform, which means that all number of cores and all bandwidths are increased by a factor of ten. The resulting platform configuration is shown in appendix B.2.2. Accordingly, the number of jobs started in the simulation are increased by a factor of ten. No emphasis has to be put on retaining the probability densities of the workloads' characteristics, since all the jobs in the validation study are characterized exactly the same, which makes the scaling trivial.

The observables obtained in the repeated validation study with the scaled platform are depicted in fig. 6.15b. Comparing them with the original validation observables shown in fig. 6.12b, which for convenience are also copied to fig. 6.15a, no significant difference



(a) Simulation **(b)** Scaled Simulation

Figure 6.15: Comparison of the validation observables predicted by the calibrated simulator (left) already shown in fig. 6.12b with the prediction obtained from the scaled simulation (right) for the validation computing platform utilised in section 6.3.3.4. In both plots, the median (points) and the 2.5- and 97.5-percentiles (whiskers) are shown for the execution times of the jobs (jobtime) separately for each executing machine in bins of the fraction of input-files read from the cache (hitrate). There is no significant difference between the two predictions observed. This has been presented for the first time in [208].

in the predicted job execution times can be observed. However, a small increase in the percentiles can be observed, which indicate a higher spread in the underlying distribution. This is to be expected due to the increased number of jobs per machine and hitrate scenario and is therefore a pure statistical effect.

It can be concluded that the presented scaling procedure conserves the dynamics of the original simulation. At the same time, for this example a boost in the execution time of the simulation of a factor of $\mathcal{O}(100)$ has been achieved on a commercially available consumer laptop. Consequently, when properly deployed, it is a valid procedure for mitigating the computational complexity of the simulator making the simulation of LSDCS feasible.

6.3.5 Large-Scale Systems

Studying the execution of current and future HEP workflows on the current and future computing infrastructure provided by the WLCG is only conceivable in simulation. The large number of components of such systems as well as their complexity challenges the achievement of a simultaneously accurate, and fast analytical performance model. As discussed in sections 6.3.3 and 6.3.4 the designed simulation model presented in sections 6.3.1 and 6.3.2 offers a reasonable trade-off between the accuracy and validity of its predictions and obtaining them in an reasonable amount of time. To showcase the ability of simulating a system with a large number of entities in the platform as well as in the simulated workload, the execution of CMS computing workloads on a hypothetical subsystem of the WLCG is studied with the simulator.

6.3.5.1 HEP Workloads

There are in general two workflows of data production processed as a prerequisite for HEP analyses, like for instance the analysis presented in chapter 5. Both typically consist of several logical processing steps. The general picture is common to all HEP collaborations. They differ, however, in their exact implementation on a collaboration by collaboration basis. In this thesis, the workflows as implemented by the CMS collaboration are studied.

First, the data recorded by the experiments needs to be processed and stored in a format that can be accessed and further processed by analysers. Hence, the digital data recorded gets reconstructed (see section 4.2). These reconstruction tasks take the digitized detector signals of an event, which correspond to an enormous amount of data, and construct high-level objects, which contain a subset of the original information. Further refining steps condense the information in the reconstructed objects, by either constructing even higher-level objects or filtering specific information irrelevant for the final physics analyses. At the same time, calibrations and corrections derived from intermediate inputs can be recycled into the processing chain and applied to the final objects meant for analysis. In CMS the original data is labelled as *RAW* data with an event size of approximately 1 MB, which gets first reconstructed into *reconstruction (RECO)* data with detailed information on the reconstructed objects resulting in an event size of roughly 3 MB [220]. Subsequent refining steps reduce the RECO data into *Analysis Object Data (AOD)* first, and further into *MINIAOD* [221] and *NANOAOD* [222] data, which reduce the event size further approximately by factors of 6 and more by one and two orders of magnitude respectively. The AOD data contains information which is necessary for deriving calibrations and corrections. The NANOAOD data is dedicated for use in most physics analyses, with most of the previously derived calibrations and corrections applied.

Second, simulation is crucial for hypothesis testing and derivation of calibrations and corrections. Therefore, using the theoretical models describing the physics of HEP collisions, events are generated using MC methods (see section 2.2). These tasks take minimal input containing the models and their configuration and generate any number of events containing the truth-level particles produced by the elaborate calculations of the physics models. The interactions of these particles with the detectors are simulated (see section 2.2.6) in an additional step, which is the computationally most expensive step of the simulation creation. After another PU mixing step (described in [40]), which superimposes the simulated events with further dedicated events, the events are fed into a simulation of the digital readout electronics. This results in simulation corresponding to the RAW data recorded by the detector but containing additional information about their origin. From here, the simulation gets handled in the same manner as the recorded data described above. The generation chain in the CMS collaboration starts with minimal configurations for the theoretical models resulting in *generator (GEN)* data. The detector simulation and PU mixing add more information from the simulation and su-

perimposed events, resulting in *simulation (SIM)* data. The *digitization (DIGI)* step creates simulated *RAW* data.

A third class of workflows is started on the resources provided by the WLCG sites by individual users. In order to run physics analyses, user jobs access the recorded and simulated processed data provided by the collaborations and use them as an input for their own applications. Since these applications' execution patterns and processed data are not fixed, user jobs span a wide range of computation and data read and write demands.

For the creation of workloads fed into the simulation study, monitoring data of the jobs executed on two WLCG sites, the tier 1 centre at KIT and the tier 2 centre at DESY, in a representative time frame, from 24th of February to the 7th of March 2023, by the CMS collaboration have been used. The jobs in the monitoring database are filtered – only successfully completed jobs are considered – and grouped into five significant classes. *Analysis* jobs cover the jobs sent by individual users. In *ReadoutSim* jobs the DIGI step of the simulation chain is covered. *Processing* jobs include several processing steps of the data reconstruction chain starting from RAW up to AOD. *Merge* jobs reduce the number of output files produced in the data reconstruction chain in order to create less input files for the next step. *Others* cover jobs that processed individual steps of the simulation and data reconstruction chain, the collection of job logs, the clean-up of logs and data and test jobs that have not occurred often enough in the monitored time frame to justify being grouped into their own class. The relative proportions of the numbers of jobs captured in each class are depicted in fig. 6.16.

The relevant characteristics of all jobs in each class are analysed. The number of requested CPU cores, the amount of requested memory, the number of input files and the amounts of data read and written by the jobs are directly extracted. For the estimation of the number of computation-operations necessary to complete a job the monitored CPU time is utilised. It is obtained by multiplying the monitored CPU time by a scaling factor obtained from the benchmark value HEP Standard Performance Evaluation Corporation 06 (HEP-SPEC06) [223] based on [224] of the corresponding site the respective job was executed on and a constant conversion factor from HEP-SPEC06 to number of operations derived from the calibration study presented in section 6.3.3.3. It is assumed that on the one hand the speed of all CPUs on the respective site is the same and constant over time. On the other hand, it is assumed that there is a direct proportionality between the number of compute operations a computer can process in a given time and the corresponding benchmark value. The cumulative distributions of the derived and directly monitored job characteristics are depicted in fig. 6.17 for each class, respectively. The corresponding quantities split by site are shown in figs. B.1 and B.2.

Simulated Workloads – The obtained distributions are used to randomly sample characteristics for job collections that make up the workloads injected into this sim-

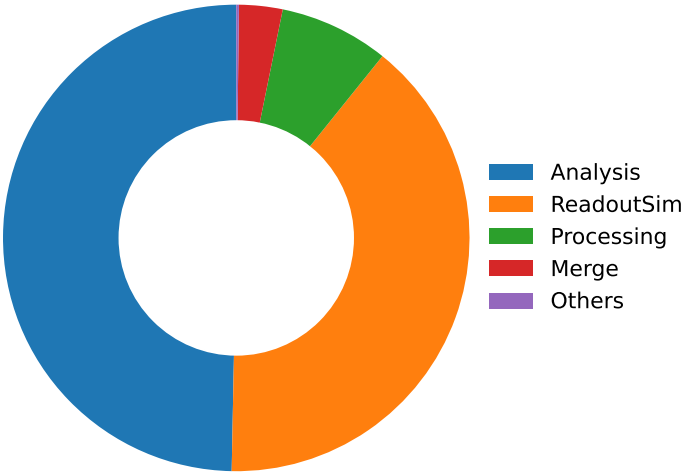


Figure 6.16: Example proportions of jobs started by the CMS collaboration on the tier 1 centre at KIT and the tier 2 centre at DESY from 24th of February to the 7th of March 2023 grouped into five classes. This plot was originally published in [208].

ulation study. Correlations between the individual characteristics are not taken into account in the sampling, which in a statistical limit will lead to a broader distribution in the phase space of characteristics. As a consequence, the observables predicted by the simulation will show a bigger spread than the ones obtained from jobs which have been executed on the real systems. For the presented studies the relative proportions of the job numbers sampled from each class are fixed to the ones obtained in the monitoring data depicted in fig. 6.16. This creates representations of workloads encountered today on the WLCG. However, future studies might explore different compositions matching future workload requirements in order to test the performance of certain architectures for these compositions.

6.3.5.2 Scenario 1: "Diskless Tier 2"

The particular study presented in the following is focused on the investigation of an aspect of an alternative architecture that has been proposed by the German HEP-computing community [225] . The idea is to remove or replace the managed disk storage operated at selected Tier 2 sites by a cache (see section 6.1.2.5). The notion behind this is saving monetary budget, since the reliable operation of a grid storage can be replaced by unmanaged storage that except for the hardware does not introduce additional costs. However, the proposal lacks further specifications of this type of cache. Additionally, it is unclear whether this would lead to losses in efficiency and performance of the respective sites. For these reasons the proposal remains purely hypothetical to date and therefore

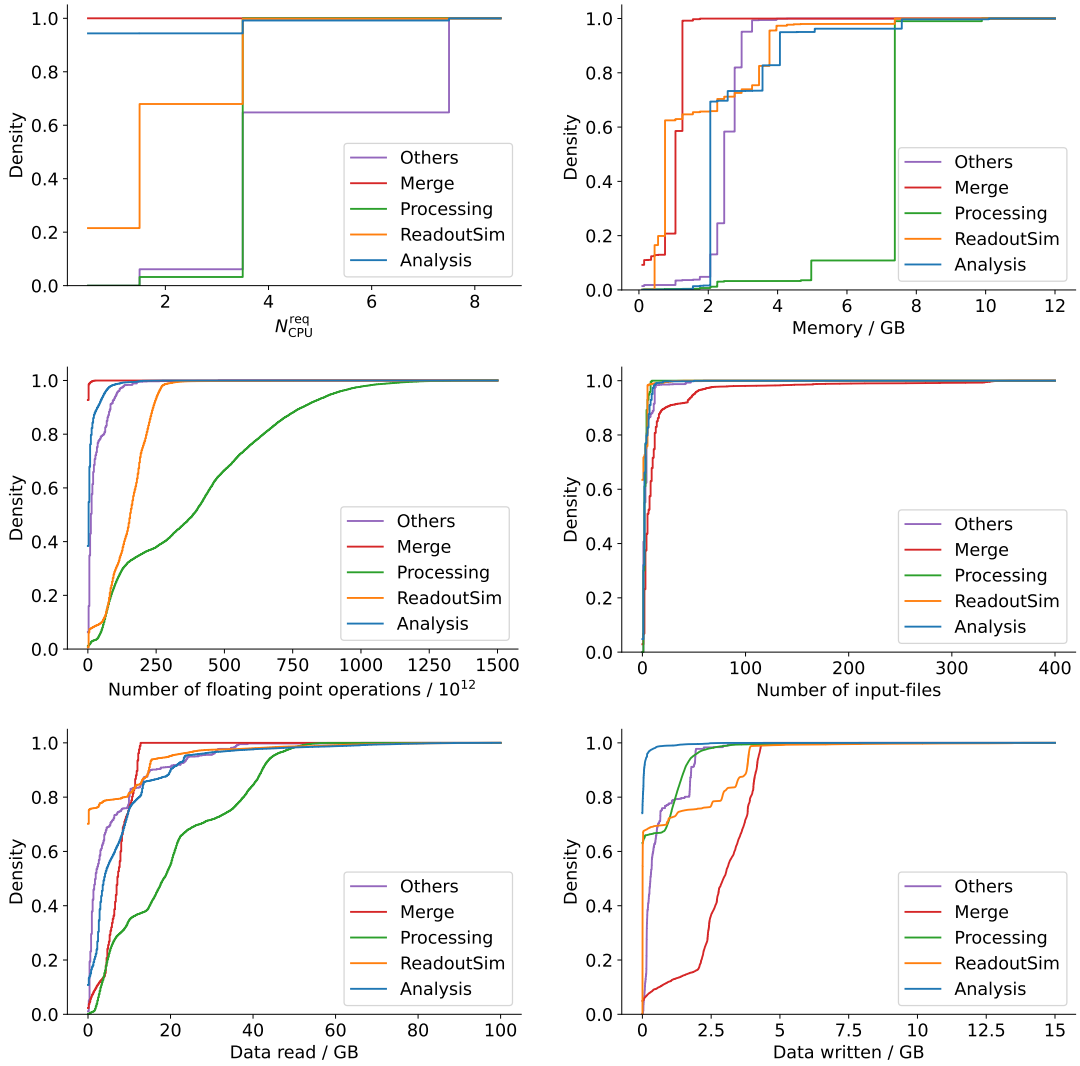


Figure 6.17: Cumulative distributions of the job characteristics for each class of jobs executed by the CMS collaboration on the tier 1 centre at KIT and the tier 2 centre at DESY from 24th of February to the 7th of March 2023. The distributions for the number of requested CPU cores $N_{\mathrm{CPU}}^{\mathrm{req}}$, the required memory, the reconstructed number of floating point operations, the number of input files and the amount of data read and written by jobs of each of the five classes is shown.

Table 6.3: Characteristics of the simulated platform used in the proof-of-concept study. It consists of a simplified WLCG sub-system with one WLCG tier 1 (Tier 1) and tier 2 site where the managed grid storage at the tier 2 centre is replaced by a cache (Tier 2').

Characteristic	Tier 1	Tier 2'	WAN
Compute	42,000 CPU cores	20,000 CPU cores	–
Storage	7 PB storage, 80 Gbit s^{-1} bandwidth	7 PB cache, 80 Gbit s^{-1} bandwidth	–
Network	2×100 Gbit s^{-1}	40 Gbit s^{-1}	100 Gbit s^{-1}

presents a perfect showcase to be studied in simulation.

It needs to be kept in mind that in such a configuration the remaining grid storage poses a single point of failure in data provisioning. The risk introduced by this is not considered. In this study it is assumed that the effect hardware failures have on operation can be neglected. Estimating the neglected effect is left for future work.

To evaluate the effect on the execution of jobs on both sites by replacing the managed storage in the tier 2 centre with a cache the job executions are studied. Since the fraction of input files present at the cache for each job is expected to have a big impact on the job execution, several simulations with preset fractions of prefetched input-files of all started jobs are started. The distributions of the selected quantities are compared for all prefetch-fraction values, also called prefetch-rate in the following, separately for all jobs executed at the same site. A prefetch-rate of value zero corresponds to a situation, where the managed storage in the tier 2 centre is expurgated without replacement. In contrast, a prefetch-rate of value one tests the nominal case of a managed grid storage at the tier 2 site since all input-files of the jobs running on this site are present. For a operational system running according to the presented idea a fraction of input-files provided by the cache between these two extremes can be achieved. However, in a realistic system this is strongly dependent on the dynamic data access patterns and the configured cache and eviction policies on the data cache. Nonetheless, for most of the time a rather lower value is expected.

Simulated Platform – The simulated system is modelled as a platform of two local networks connected by a single WAN link, as depicted in fig. 6.18. Each local network represents a WLCG site. The storage resource at the tier 2 site is operated as a data cache, which would disqualify the site from being a WLCG tier 2 site (see section 6.1.1.1). Therefore, it will be labelled as "Tier 2'" in the following. The exact characteristics of the two sites' representation in the simulation are based on the tier 1 centre at KIT and the tier 2 centre at DESY. Their approximate characteristics assumed in this thesis are summarized in table 6.3.

This simulation platform aims to approximate the behaviour of its real-world coun-

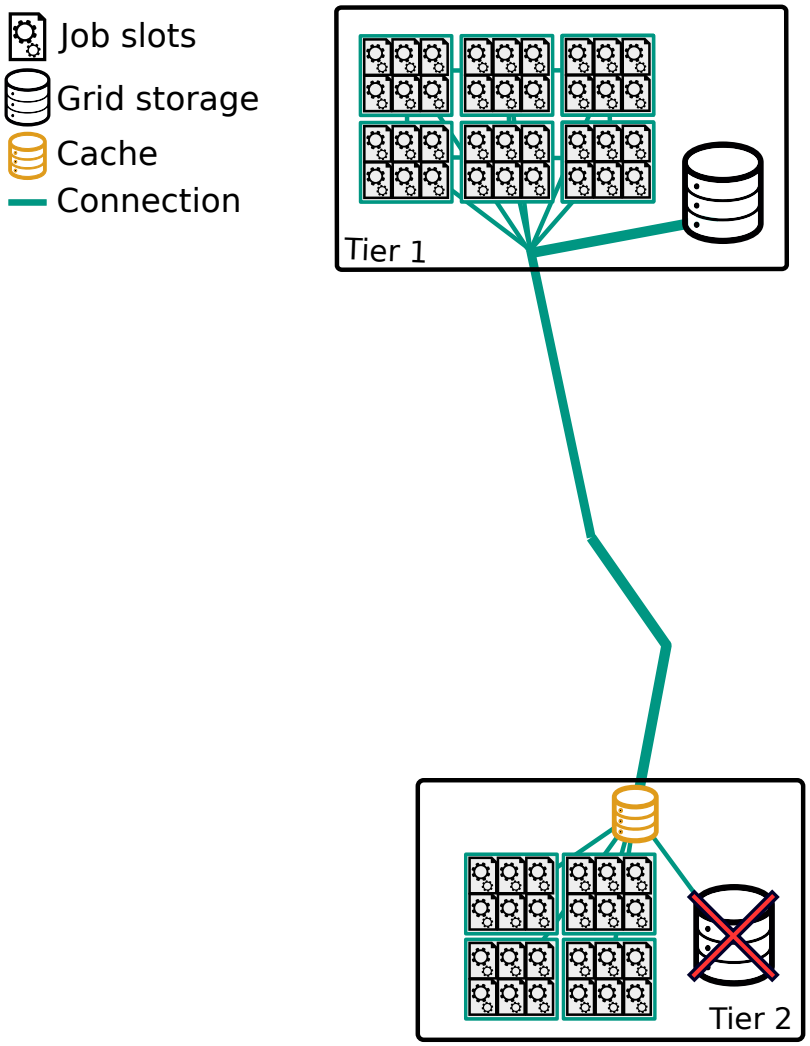


Figure 6.18: Sketch of a simplified WLCG sub-system with one WLCG tier 1 (Tier 1) and tier 2 site where the managed grid storage at the tier 2 centre is replaced by a cache (Tier 2). Each site is its own local network of interconnected storage and compute resources. The local networks are connected by a WAN link.

terpart. Although both sites are at first order isolated, the WAN in between is shared not only by the two modelled centres but is also part of the bigger network embedding the modelled system. Therefore, it is expected that loads on the shared network links introduced by activities executed on remote sites but transferring data over the links in the real system impact the execution of jobs on the tier 2 centre. Additionally, in case the remote activities put load on the grid storage on the tier 1 site an impact on the execution of the local jobs is expected. Neither of these impacts are modelled for this platform. Consequently, the derived observables from the simulation are expected to change when including these impacts. They can be estimated without a detailed simulation of the remote sites by reducing the maximum available bandwidth on the corresponding links and storage services in the simulation. Nevertheless, this would require exact knowledge about the dynamic network traffic patterns at each component, which poses an additional challenge. Alternatively, the impacts can be approximated by reducing the bandwidths randomly. However, the question remains which probability distribution is to chose for this random reduction. It is left for future studies to find a solution for these open challenges.

For the simulation's predictions to be realistic, the calibrations derived in section 6.3.3.3 are applied to the simulated platform. This includes an increase of all bandwidths by approximately 15%, and a decrease of the CPU speeds in simulation by approximately 20%. However, the estimation of the number of operations processed for each job in this showcase scenario is different from the procedure in section 6.3.3 because the derived calibration of the CPU speed might not be suitable for this scenario. Therefore, the predicted job execution distributions in simulation with a prefetch-rate of one are compared to the ones in the gathered job monitoring data. From this a supplementary calibration of the CPU speeds in the simulation is derived. A minor correction of the CPU speeds on top of the calibrated values of $\mathcal{O}(1\%)$ emerges, which is expected to be small compared to the effect introduced by the sampling of the job characteristics or the neglecting of remote influences on the WAN. The simulation of a platform with $\mathcal{O}(10000)$ simultaneously active jobs is not feasible, as discussed in section 6.3.4. Therefore, the platform is scaled down by a factor of 100 to decrease the runtime of the simulation. The exact SimGrid platform configuration resulting from the calibration and scaling is shown in appendix B.2.3.

Results – The total job execution time, the time spent transferring input and output data, the time spent processing, and the fraction of time spent processing over the total job execution time, called the CPU efficiency, are reconstructed from the monitored quantities in simulation. For studying the whole collection of executed jobs the empirical probability distributions are studied. Therefore, the distributions for each prefetch-rate value are visualized as a box plot showing the median, the 25- and 75-percentiles, and the value of the entry with the smallest (biggest) value within the 25-percentile minus (plus) 1.5 times the inter percentile range. The selected quantities are shown in fig. 6.19.

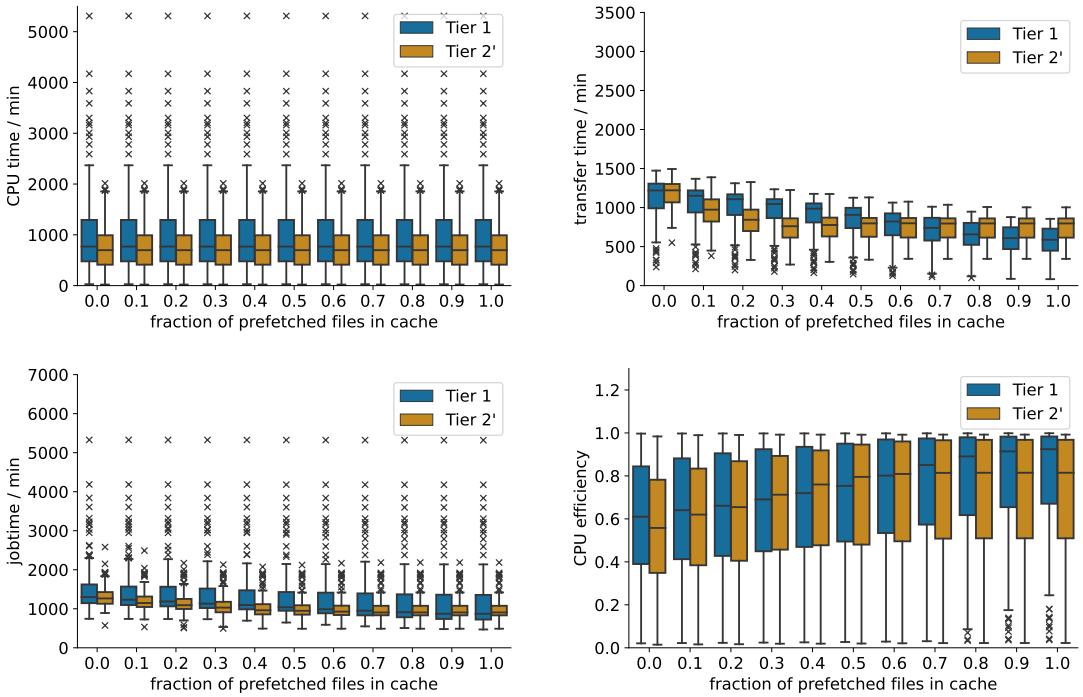


Figure 6.19: The simulated observables for the job execution on the simulated platform consisting of a tier 1 and a "diskless" tier 2 site as a function of the prefetch-rate. The distributions for the executed jobs per site of the time spent transferring input and output data (transfer time), the time spent processing (CPU time), the total job execution time (jobtime) and the CPU efficiency are visualized. For each prefetch-rate value the median (line through the box), the 25- and 75-percentiles (lower and upper end of the box), and the value of the entry with the smallest (biggest) value within the 25-percentile minus (plus) 1.5 times the inter percentile range (whiskers) is shown. Individual outliers, which reconstructed quantity is not within the range are plotted as crosses. Similar plots for the transfer time and CPU efficiency have been presented for the first time in [208].

It can be observed that for all prefetch-rate values the time spent in processing is the same. This shows that the pure processing is invariant of the fraction of prefetched files matching the expectation. By contrast, the time spent in transferring is dependent on the prefetch-rate. As expected, it can be observed that for higher prefetch-rate values the total time spent by the jobs executed at Tier 2'transferring input and output data decreases. The decrease flattens at prefetch-rate values $\gtrapprox 0.5$. This indicates that the lower local network bandwidth at Tier 2'throttles the execution of the jobs reading a substantial fraction of input data from the cache. However, at the tier 1 site a gradual decrease in the transfer time can be observed. Since with a higher fraction of jobs at the tier 2 site reading input data from the cache instead of from the managed grid storage at the tier 1 site bandwidth on the grid storage is freed, which benefits the execution of the jobs running on the tier 1. Consequently, the same effect can also be observed indirectly in the job execution time (jobtime). For all jobs the jobtime is reduced with higher prefetch-rates because of the faster data access. But the decrease for jobs running on Tier 2' flattens for prefetch-rates $\gtrapprox 0.5$. Hence, job execution on both sites benefits from the faster data access.

This can also be observed in terms of CPU efficiency. For a prefetch rate of 1.0 by construction the CPU efficiency corresponds to the nominal case of a separate grid storage on each site. Comparing the simulated CPU efficiencies for lower prefetch rates a significant drop in the efficiency of the jobs executed on both sites can be observed due to the increasing load on the tier 1 grid storage that has to substitute the missing grid storage at Tier 2'. Since it is not designed to cover the input and output bandwidth requirements posed by a number of jobs concurrently running on two sites sharing the storage resource it becomes the limiting component in the execution of the workloads. For a realistic platform where the grid storage at Tier 2' is replaced by a cache a significant drop in the CPU efficiency up to 20% is expected. This is unfavourable for the operation of both sites. Consequently, the data cache at the tier 2 site as a replacement of the grid storage is only reasonable with simultaneous performance improvements of the grid storage at the tier 1 so that it is able to serve the increased data access requirements in this scenario.

Repeating the same simulation with an increased bandwidth of the remote storage at the Tier 1 by 25% reveals the expected improvement. In fig. 6.20 the resulting traces of the job execution in simulation are shown. In comparison to the nominal case (see fig. 6.19), the job execution on both sites is more efficient even for small fractions of input files provided by the cache and the execution of the jobs at the tier 2 site reaches a full utilisation of the bandwidth of the connection between the sites for smaller prefetch values. Furthermore, due to the upgrade of the grid storage's bandwidth at the tier 1 site the processing of the jobs is in general more efficient for all prefetch values. This indicates, that the jobs executed at the Tier 1 profit from the faster data transfers enabled by the more performant grid storage. Moreover, already for prefetch values $\gtrapprox 0.1$ maximum efficiency of the job execution at the tier 2 site for this architecture is reached. Consequently, a grid storage with higher bandwidth serving multiple sites improves on

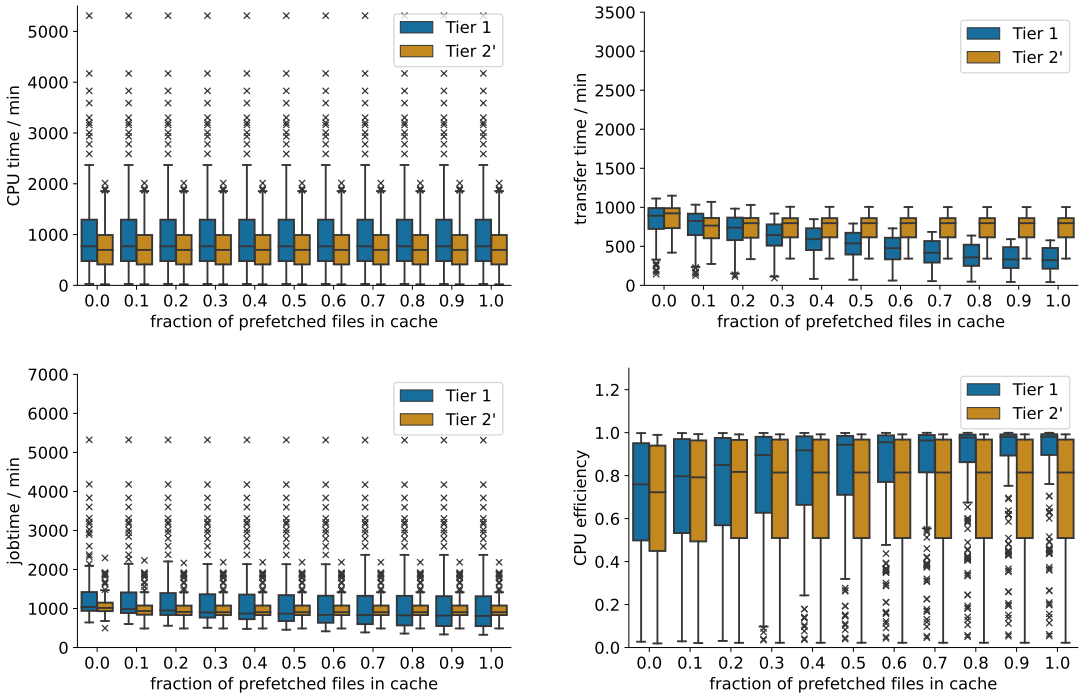


Figure 6.20: The simulated observables for the job execution on the simulated platform consisting of a tier 1 with increased grid storage bandwidth by 25% and a "diskless" tier 2 site as a function of the prefetch-rate. The same distributions as in fig. 6.19 are shown for comparison. The upgraded bandwidth of the grid storage serving both sites improves the efficiency of the executed jobs significantly, even for small prefetch values.

one hand the job execution in general and on the other hand reduces the reliance on high fractions of input data served by the cache.

6.3.5.3 Scenario 2: Replacement of a Tier 2 by an HPC Centre

A natural follow-up to the idea discussed in section 6.3.5.2 is the replacement of the whole tier 2 site by alternate resources. In the German HEP-computing community the idea to outsource the operation of dedicated Tier 2s to national supercomputers, so-called high-performance computing (HPC) centres, has become popular [225]. Indeed, the number of CPU cores provided by HPC centres typically exceeds the available number on most German tier 2 sites. However, in contrast to WLCG tier 2 sites HPC centres are not designed for the execution of data intensive tasks (see sections 6.1.1.1 and 6.1.1.2). This manifests for instance in a lower bandwidth connection by factors of $\mathcal{O}(10)$ to the WAN of HPC centres. Additionally, due to the event-based nature of HEP workloads, as discussed in section 6.1.1.2, they do not profit from the internal high bandwidth links in the HPC centres since the transfers of the input data are throttled by the remote connection. Consequently, the execution of jobs on HPC centres is expected to be less efficient than on regular tier 2 sites, since the latter are specifically optimized in terms of data throughput for these kinds of workloads. Especially the lack of a high bandwidth remote network connection is expected to be harmful in HEP data processing scenarios on HPCs, since input data for the processing has to be transferred from remote. Therefore, the addition of a data cache in close vicinity to the HPC site tethered to the HPC's compute resources with a high bandwidth is expected to improve the execution of HEP jobs significantly.

A study of the performance of such a system presents an additional scenario perfectly suited for being investigated in simulation.

Simulated Platform – The discussed platform is modelled using the platform presented in section 6.3.5.2 as a base. In this scenario, it is assumed that the characteristics of both sites remain unchanged. This simplifies the comparison with the results obtained in simulation of the previous scenario. Consequently in this context, the HPC centre replacing the tier 2 site (also labelled as Tier 2' in the following) shares exactly the same characteristics. However, the bandwidth of the link between the two sites is reduced by a factor of 10.

Typically, modern HPC centres are equipped with a fast internal network between its individual worker nodes to enable fast execution of multi-node applications exchanging data between nodes. However, not all types of applications and transfer protocols can utilise these fast bandwidths. It remains an open question whether and how HEP applications can profit from the fast internal network. Therefore, the effect of an increase in the local links' bandwidths in the simulation's HPC centre representation is not studied. Nevertheless, such a study would require only a small change in the platform configuration of the simulation and can therefore be achieved with minimal effort.

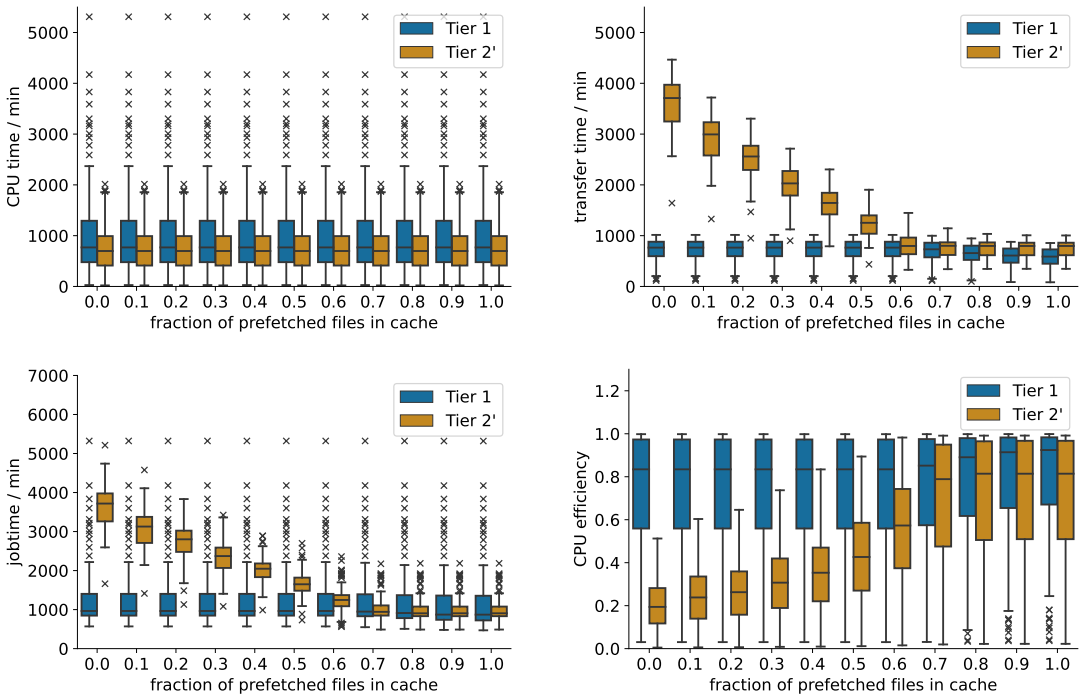


Figure 6.21: The simulated observables for the job execution on the simulated platform consisting of a tier 1 and an HPC centre replacing a tier 2 site as a function of the prefetch-rate. For comparison the same visualizations of the distributions as depicted in fig. 6.19 are shown. Since the HPC centre's connection is worse than that of a customized Tier 2, the observed effects on the job executions at the HPC are more pronounced. A similar plot for the CPU efficiency has been presented for the first time in [208].

Results – The studies performed in section 6.3.5.2 are repeated with the adjusted platform. The same workload as in the previous scenario is simulated for all prefetch-rate values. For comparison, the same visualizations of the distributions as depicted in the previous scenario are shown in fig. 6.21.

As expected, the time spent by the jobs in processing their workloads remains unchanged for all prefetch rates and is not affected by the platform changes made for this scenario. The transfer of input and output data however differs significantly compared to the previous scenario. Due to the smaller bandwidth of the link connecting both sites the decrease in the transfer times of the jobs with higher prefetch-rate values is more pronounced. As expected, the effect on the job execution due to a lack of an operational data cache leads to a decline in the data access performance of the jobs executed on this scenario's Tier 2'. The limited bandwidth for the jobs running on the HPC and accessing files present at the tier 1 site's grid storage leads to a throttling of their execution. Jobs being executed on the tier 1 site however benefit from this. Since their local grid storage is not at full capacity the bandwidth can be utilised by them. This leads to the

observed flat evolution with the prefetch-rate in the transfer times of the jobs executed on that site. The same effects can also be observed in the jobs' execution times and CPU efficiencies. While the jobs executed on the tier 1 site are only slightly effected by the prefetch-rate the efficiency of the execution of the jobs on Tier 2' is significantly increased for high prefetch rates.

The nominal case of a dedicated tier 2 site, corresponding approximately to the case of a prefetch-rate value of 1.0, leads to the best performance. However, if the replacement of a dedicated tier 2 site is insisted on, the inclusion of a data cache significantly boosts the performance of the execution of HEP jobs on the HPC centre. The execution of jobs on the tier 1 site remains insignificantly affected. Nonetheless, in a realistic scenario where fractions of input data provided by the data cache are expected to be low the achievable efficiencies remain smaller than in the nominal case, which remains an unsolved challenge for the operation of the HPC centre. Due to the data-intensive nature of HEP jobs this can only be solved by increasing the bandwidth to the HPC centre, unless one can figure out how to predict the data access patterns of HEP jobs in such detail that fractions of input data provided by the data cache above approximately 70% are achieved. The solution to the latter challenge is left for future studies.

CHAPTER 7

Conclusions

Within the scope of this thesis a measurement of the differential cross sections for the production of oppositely charged muon pairs with invariant mass close to the mass of a Z boson in association with at least one jet in proton-proton collisions at a center-of-mass energy of 13 TeV is presented in chapter 5. For short, the analysed topology in this work is labelled as Z boson plus jet (Z+jet). The measurement is corrected for experimental effects introduced by the detector and the subsequent reconstruction of the measured signals using the unfolding technique presented in section 5.5. This allows for a direct comparison with theoretical predictions, skipping the expensive folding of predicted results with simulations of the detector and reconstruction, and evaluating the predictive power of various theoretical models. In exchange for this straightforward interpretability, an additional uncertainty related to the unfolding technique that scales with the statistical power of the utilised simulation is accounted.

The analysed dataset in this thesis recorded by the Compact Muon Solenoid (CMS) collaboration corresponds to an unprecedented amount of events. This leads to approximately 10.5 million recorded events passing the selections of the presented analysis allowing to perform the precise measurement of the differential cross sections in 264 individual analysis bins. Even for high transverse momenta of the dimuon system p_T^{Z} and high rapidities y_b and y^*, where the population of selected events is expected to be small, statistical uncertainties smaller than 10% are achieved. Systematic uncertainties dominate for $p_T^{\mathrm{Z}} \lesssim 80\,\mathrm{GeV}$, specifically the ones originating in the calibration of the jet energy scale. At intermediate p_T^{Z} around 100 GeV, most contributions to the total uncertainties originate from the calibration of the jet energy scale and estimation of backgrounds. In spite of using an unprecedented amount of Z+jet events leading to the best achievable statistical power statistical uncertainties are dominating for $p_T^{\mathrm{Z}} \gtrsim 150\,\mathrm{GeV}$. Since the simulated sample used in the unfolding matches the expected statistical power in data the corresponding unfolding uncertainty is of the same order of magnitude. As future operations of the Large Hadron Collider (LHC), and its future upgrade the High Luminosity Large Hadron Collider (HL-LHC), are expected to increase the amount of data by a factor of ten, the results in these statistically limited regions will improve in this regard.

However, the already achieved precision in the presented measurement allows for significant comparisons of the measured cross sections with theoretical predictions as, presented in section 5.7, and rule out inapt predictions. The predictions utilised in the comparisons are made with MadGraph5_aMC@NLO and Pythia 8 event generators and merge fixed order calculations at different jet multiplicities at leading order (LO) and next-to-leading order (NLO) accuracy in perturbative quantum chromodynamics (QCD) with the MLM and FxFx merging methods, respectively. While both methods approximate predictions at high perturbative orders, the latter provides the best approximation to a prediction at next-to-next-to-leading order (NNLO) accuracy in perturbative QCD. While the former fails to describe the data, predictions made by the latter describe the differential cross sections in p_T^{Z} well. For the two analysed rapidities a systematic trend in y^*, but not in y_b, is observed. However, the trend is just at the edge of compatibility given the assigned uncertainty on the measurement and the predictions.

This indicated tension between the measured cross sections and the predictions might be either confirmed or resolved by predictions at full NNLO accuracy with correspondingly smaller uncertainty. State of the art NNLO perturbative predictions for the analysed observables, however, cannot be produced with modelling of non-perturbative QCD effects that are expected to add significant contributions in low p_T^{Z} and high rapidity regions. Therefore, the contributions of non-perturbative QCD effects are estimated in section 5.3.2.2 using the Herwig event generator. As a result, the cross sections at $p_T^{\mathrm{Z}} \lesssim 50\,\mathrm{GeV}$ are significantly affected by non-perturbative QCD effects. Additionally, a systematic trend with y^*, but not with y_b, in the size of the effects is observed indicating a feature of the analysed phase space, which may be further investigated in future studies. For higher order predictions the non-perturbative QCD effects diminish. For $p_T^{\mathrm{Z}} \gtrsim 100\,\mathrm{GeV}$ the observed effects are aligned with zero. Since the low p_T^{Z} regions are also dominated by systematic uncertainties, interpretations of the perturbative modelling utilising the results of the presented measurement, for instance improved PDF fits, can profit from excluding the affected bins. Nonetheless, these bins provide sensitivity to the non-perturbative modelling motivating dedicated future studies on the effects of hadronization and underlying event on the production of Z+jet events at the LHC.

All the presented studies are enabled by large-scale distributed computing infrastructures providing the storage capacity for the data and processing power. The infrastructure usable by individual scientists and the LHC collaborations, consists of a global coalition of computing sites that are combined to form the Worldwide LHC Computing Grid (WLCG). All kinds of reconstruction and simulation tasks needed for calibration and analysis, like the ones presented in this work, are executed on the WLCG and affiliated sites. The processing and storage demands for this thesis alone analysing the reconstructed data measured in three years of data-taking sum up to orders of 100 kCPUh and more than 200 TB, respectively, and the event generations add additional processing demands at the order of 100 kCPUh. In this estimate, the previous analysis and calibration steps performed within and the simulated samples generated by the CMS collaboration are not included. They are expected to add computing demands of an

order of magnitude larger than for this analysis' purpose only. Similar demands by the other three collaborations at the LHC are estimated.

With the start of the HL-LHC, anticipated for the end of the decade, the data rate is expected to increase tenfold. This will enable future precision analyses, but also pose record computing demands on the shared infrastructure rendering an optimized usage in terms of efficiency necessary. As a consequence of the near-term start of the HL-LHC, the computing infrastructure of the future has to be designed now. For the identification of optimized infrastructures, however, performance modelling and simulation studies are needed. In the years since the initial operation, the infrastructure has become more heterogeneous and complex following contemporary trends and developments in the computing hardware. These developments are not covered by the models and tools used in the original design of the WLCG.

Therefore, as part of this thesis, a new approach for the performance modelling of high energy physics (HEP) workflows on distributed computing infrastructures like the WLCG has been developed and presented in chapter 6. For this purpose, new models have been developed and implemented into a simulator tool aiming to accurately describe the execution of modern and future workloads on distributed computing infrastructures, as described in section 6.3. This tool and its underlying models are calibrated and validated exhibiting the potential to produce accurate predictions, presented in section 6.3.3. In a study of infrastructure candidates inspired by a proposal of the German HEP computing community the performance of such systems is predicted in section 6.3.5. Using the simulation tool, the advantages and disadvantages of such a system in terms of execution efficiency and performance are identified. Furthermore, based on the predictions made, improvements to the initial design are postulated, and their impact is validated with another simulation study. This demonstrates the applicability of the developed tool for studying complex distributed computing infrastructure designs, providing a proof of principle for future design studies adopting the established models in view of the unprecedented computing demands in the HL-LHC era.

APPENDIX A

Supplementary Analysis Material

A.1 Derivation of NP-Corrections

A.1.1 Example Herwig Configuration File

The following configuration file is used for the generation of the datasets using MadGraph5_aMC@NLO [97, 98] for the generation of the hard scattering at LO accuracy in QCD interfaced to Herwig [27, 28]:

```
# -*- ThePEG-repository -*-

##################################################
## Herwig/Matchbox example input file
##################################################

############################
##  Setup the MonacoSampler.
############################

read snippets/MonacoSampler.in

cd /Herwig/Samplers
# Perform the addaption in the read/integrate step with:
set MonacoSampler:NIterations 7
set MonacoSampler:InitialPoints 40000
set MonacoSampler:EnhancementFactor 1.3

##################################################
## Collider type
##################################################
read snippets/Matchbox.in
read snippets/PPCollider.in
```

```
##################################################
## Beam energy sqrt(s)
##################################################

cd /Herwig/EventHandlers
set EventHandler:LuminosityFunction:Energy 13000*GeV

##################################################
## Process selection
##################################################

## Note that event generation may fail if no matching
## matrix element has been found.  Coupling orders are
## with respect to the Born process, i.e. NLO QCD does
## not require an additional power of alphas.

## Model assumptions
read Matchbox/StandardModelLike.in
#read Matchbox/DiagonalCKM.in

## Set the order of the couplings
cd /Herwig/MatrixElements/Matchbox
set Factory:OrderInAlphaS 1
set Factory:OrderInAlphaEW 2

## Select the process
## You may use identifiers such as p, pbar, j, l, mu+,
## h0 etc.
do Factory:Process p p -> mu+ mu- j

## Special settings required for on-shell production of
## unstable particles enable for on-shell top production
# read Matchbox/OnShellTopProduction.in
## enable for on-shell W, Z or h production
# read Matchbox/OnShellWProduction.in
# read Matchbox/OnShellZProduction.in
# read Matchbox/OnShellHProduction.in
# Special settings for the VBF approximation
# read Matchbox/VBFDiagramsOnly.in

##################################################
## Matrix element library selection
##################################################
```

```
## Select a generic tree/loop combination or a
## specialized NLO package

# read Matchbox/MadGraph-GoSam.in
# read Matchbox/MadGraph-MadGraph.in
# read Matchbox/MadGraph-NJet.in
# read Matchbox/MadGraph-OpenLoops.in
# read Matchbox/HJets.in
# read Matchbox/VBFNLO.in

## Uncomment this to use ggh effective couplings
## currently only supported by MadGraph-GoSam

# read Matchbox/HiggsEffective.in

#######################################################
## Cut selection
## See the documentation for more options
#######################################################
cd /Herwig/Cuts/
set ChargedLeptonPairMassCut:MinMass 60*GeV
set ChargedLeptonPairMassCut:MaxMass 120*GeV

## cuts on additional jets

read Matchbox/DefaultPPJets.in

insert JetCuts:JetRegions 0 FirstJet
# insert JetCuts:JetRegions 1 SecondJet
# insert JetCuts:JetRegions 2 ThirdJet
# insert JetCuts:JetRegions 3 FourthJet

#######################################################
## Scale choice
## See the documentation for more options
#######################################################

cd /Herwig/MatrixElements/Matchbox
set Factory:ScaleChoice /Herwig/MatrixElements/Matchbox/
Scales/LeptonPairMassScale

#######################################################
## Matching and shower selection
## Please also see flavour scheme settings
```

```
## towards the end of the input file.
##################################################################

#read Matchbox/MCatNLO-DefaultShower.in
# read Matchbox/Powheg-DefaultShower.in
## use for strict LO/NLO comparisons
read Matchbox/MCatLO-DefaultShower.in
## use for improved LO showering
# read Matchbox/LO-DefaultShower.in

# read Matchbox/MCatNLO-DipoleShower.in
# read Matchbox/Powheg-DipoleShower.in
## use for strict LO/NLO comparisons
# read Matchbox/MCatLO-DipoleShower.in
## use for improved LO showering
# read Matchbox/LO-DipoleShower.in

# read Matchbox/NLO-NoShower.in
# read Matchbox/LO-NoShower.in

##################################################################
## Scale uncertainties
##################################################################

read Matchbox/MuDown.in
read Matchbox/MuUp.in

##################################################################
## Shower scale uncertainties
##################################################################

read Matchbox/MuQDown.in
read Matchbox/MuQUp.in

##################################################################
## PDF choice
##################################################################

read Matchbox/FiveFlavourScheme.in
## required for dipole shower and fixed order in
## five flavour scheme
# read Matchbox/FiveFlavourNoBMassScheme.in
read Matchbox/CT14.in
# read Matchbox/MMHT2014.in
```

```
##################################################
## Analyses
##################################################

cd /Herwig/Analysis
## Write HepMC events. Modify the PrintEvent interface
## for your needs.
set /Herwig/Analysis/HepMC:PrintEvent 100000
set /Herwig/Analysis/HepMC:Format GenEvent
set /Herwig/Analysis/HepMC:Units GeV_mm
insert /Herwig/Generators/EventGenerator:AnalysisHandlers 0
/Herwig/Analysis/HepMC

##################################################
## Save the generator
##################################################

do /Herwig/MatrixElements/Matchbox/Factory:ProductionMode

set /Herwig/Generators/EventGenerator:IntermediateOutput Yes

cd /Herwig/Generators
saverun LHC-LO-ZplusJet EventGenerator
```

For the sample produced at NLO accuracy in QCD a similar configuration file is used. In the NLO production the number of sampling points is increased by 50% to account for the more complex phase space integration. Additionally, to set the NLO production active the line `read Matchbox/MCatLO-DefaultShower.in` is changed to `read Matchbox/MCatNLO-DefaultShower.in`.

Parts of the generation modules in Herwig are turned off for the partial generation chain passing the following setup file:

```
1 ##################################################
2 ## ShowerHandler(s)
3 ##################################################
4
5 ## Switches for turning generation steps off and on
6 cd /Herwig/EventHandlers
7 set EventHandler:CascadeHandler:MPIHandler NULL
8 set EventHandler:DecayHandler NULL
9 set EventHandler:HadronizationHandler NULL
```

A.1.2 Rivet Routine

```
// -*- C++ -*-
#include "Rivet/Analysis.hh"
#include "Rivet/Projections/FinalState.hh"
#include "Rivet/Projections/ChargedFinalState.hh"
#include "Rivet/Projections/PromptFinalState.hh"
#include "Rivet/Projections/VetoedFinalState.hh"
#include "Rivet/Projections/DressedLeptons.hh"
#include "Rivet/Projections/FastJets.hh"
#include "Rivet/Projections/JetAlg.hh"
#include "Rivet/Projections/MissingMomentum.hh"

namespace Rivet {


    /// @brief Add a short analysis description here
    class ZplusJet_3 : public Analysis {
    public:

    /// Constructor
    DEFAULT_RIVET_ANALYSIS_CTOR(ZplusJet_3);


    /// @name Analysis methods
    ///@{

    /// Book histograms and initialise projections before the
    /// run
    void init() {

        MSG_INFO(
        "Analysis cuts: \n"
        << "\tminimum jet pt for jet definition: "
        << _jetpt << "\n"
        << "\tminimum jet1 pt: " << _minjet1pt << "\n"
        << "\tjet rapidity cut: " << _maxabsjetrap << "\n"
        << "\tmaximum number of leptons: "
        << _maxnleptons << "\n"
        << "\tminimum lepton pt: " << _minleptonpt << "\n"
        << "\tlepton eta cut: " << _maxleptoneta << "\n"
        << "\tZ-mass window +/-: " << _massdiff << "\n"
        << "\tminimum Z pt: " << _minptZ << "\n"
        );
```

```
// Initialise and register projections

// The basic final-state projection:
// all final-state particles within
// the given eta acceptance
const FinalState fs(
    Cuts::abseta < 5. && Cuts::pT > 100*MeV
);
//const ChargedFinalState cfs(fs);


// The final-state particles declared above are
// clustered using FastJet with the
// anti-kT algorithm and a jet-radius parameter 0.4
// neutrinos are excluded from the clustering
FastJets jetfsak4(
    fs,
    FastJets::ANTIKT, 0.4,
    JetAlg::Muons::ALL, JetAlg::Invisibles::NONE
);
declare(jetfsak4, "jetsAK4");
FastJets jetfsak8(
    fs,
    FastJets::ANTIKT, 0.8,
    JetAlg::Muons::ALL, JetAlg::Invisibles::NONE
);
declare(jetfsak8, "jetsAK8");

// FinalState of prompt photons and bare muons
// and electrons in the event
PromptFinalState photons(Cuts::abspid == PID::PHOTON);
PromptFinalState bare_leps(
    Cuts::abspid == PID::MUON
        || Cuts::abspid == PID::ELECTRON
);

// Dress the prompt bare leptons with prompt photons
// within dR < 0.1,
// and apply some fiducial cuts on the dressed leptons
Cut lepton_cuts = Cuts::abseta < _maxleptoneta
    && Cuts::pT > _minleptonpt;
DressedLeptons dressed_leps(
    photons, bare_leps, 0.1, lepton_cuts
```

```
);
declare(dressed_leps, "leptons");

// Missing momentum
/// Out of acceptance particles treat as invisible
VetoedFinalState fs_onlyinacc(
    fs,
    (Cuts::abspid == PID::MUON && Cuts::abseta > 2.4)
    ||(Cuts::abspid == PID::PHOTON
    && Cuts::abseta > 3.0)
    ||(Cuts::abspid == PID::ELECTRON
    && Cuts::abseta > 3.0)
);
declare(MissingMomentum(fs_onlyinacc), "MET");

// Book histograms
// specify custom binning
/// Book histograms with variable bin size

vector<double> binedges_Ystar = {
    0.0, 0.5, 1.0, 1.5, 2.0
};
vector<double> binedges_Yboost = {
    0.0, 0.5, 1.0, 1.5, 2.0
};

vector<double> binedges_ZPt;

for(string _jettype: {"AK4","AK8"}){
book(
    _h["NJets"+_jettype], "NJets"+_jettype,
    11, -0.5, 10.5
);
for(auto _ystar: binedges_Ystar){
    for(auto _yboost: binedges_Yboost){
    if(_ystar + _yboost > 2.) continue;
    // extreme bin
    if(_ystar>=2.0 && _yboost<0.5){
        binedges_ZPt = {25., 30., 40., 50.,
        70., 90., 110., 150., 250.};
    }
    // central bins
    else if((_ystar<0.5 && _yboost<2.)
        || (_ystar<1. && _yboost<1.5)
```

```
            || (_ystar<1.5 && _yboost<1.)){
            binedges_ZPt = {25., 30., 35., 40., 50.,
            60., 70., 80., 90., 100., 110.,
            130., 150., 170., 190., 220.,
            250., 400., 1000.};
        }
        // edge bins
        else {
            binedges_ZPt = {25., 30., 35., 40., 45., 50.,
            60., 70., 80., 90., 100., 110.,
            130., 150., 170., 190.,
            250., 1000.};
        }

        string _hist_ZPt_ident = "ZPt"+_jettype
            +"Ys"+to_string(_ystar)+"Yb"
            +to_string(_yboost);
        string _hist_ZPt_name = _hist_ZPt_ident;

        book(
            _h[_hist_ZPt_ident], _hist_ZPt_name,
            binedges_ZPt
        );

        }
    }
    }

    MSG_INFO(
    "Booked " << _h.size() << " histograms"
    );
    if (getLog().isActive(Log::DEBUG)) {
    MSG_DEBUG("Histograms:");
    for (auto h: _h) {
        MSG_DEBUG("\t" << h.first);
    }
    }
}


/// Perform the per-event analysis
void analyze(const Event& event) {

    // Retrieve dressed leptons, sorted by pT
```

```
vector<DressedLepton> leptons = apply<DressedLeptons>(
    event, "leptons"
).dressedLeptons();

// discard events with less than two and
// more than maximum number of leptons
if (leptons.size() < 2) vetoEvent;
if (leptons.size() > _maxnleptons) vetoEvent;
MSG_DEBUG("Found " << leptons.size() << " leptons");
for (auto lep: leptons) {
MSG_DEBUG("\tlepton pt: " << lep.pT());
MSG_DEBUG("\tlepton y: " << lep.rap());
}

// Retrieve clustered jets, sorted by pT,
// with a minimum pT cut
map<string,Jets> _jetcollections;
_jetcollections["AK4"] = apply<FastJets>(
    event, "jetsAK4"
).jetsByPt(
    Cuts::absrap < _maxabsjetrap && Cuts::pT > _jetpt
);
_jetcollections["AK8"] = apply<FastJets>(
    event, "jetsAK8"
).jetsByPt(
    Cuts::absrap < _maxabsjetrap && Cuts::pT > _jetpt
);

// Require at least one jet in any jet collection
// with a minimum pT
bool jet1pass = false;

set<string> _jetcollectionstoerase;

for (auto& jets: _jetcollections) {
// Remove all jets within dR < 0.3 of a dressed lepton
idiscardIfAnyDeltaRLess(
    jets.second, leptons, _lepCleaningDeltaR
);
MSG_DEBUG("After lepton cleaning jet multiplicity "
    << jets.first << "= " << jets.second.size());
for (auto jet: jets.second) {
    MSG_DEBUG("\tjet pt: " << jet.pT()/GeV);
    MSG_DEBUG("\tjet y: " << jet.rap());
```

```
}

// Require at least one hard jet
if (!jets.second.empty()) {
    if (jets.second.at(0).pT() > _minjet1pt) {
    MSG_DEBUG(
        "Hardest " << jets.first <<
        " jet pt: " << jets.second.at(0).pT()
    );
    jet1pass = true;
    } else {
    _jetcollectionstoerase.insert(jets.first);
    }
}
else {
    _jetcollectionstoerase.insert(jets.first);
}
}

if (!(jet1pass)) vetoEvent;

for (string c: _jetcollectionstoerase) {
_jetcollections.erase(c);
}
MSG_DEBUG("Remaining jet collections:");
if (getLog().isActive(Log::DEBUG)) {
for (auto jc: _jetcollections) {
    MSG_DEBUG("\t" << jc.first);
}
}

// Require at least two opposite sign leptons
// compatible with Z-boson mass and keep the pair
// closest to Zboson mass
bool bosoncandidateexists = false;
double massdiff = _massdiff;
DressedLepton muon = leptons.at(0);
DressedLepton antimuon = leptons.at(0);

for (unsigned int it = 1; it < leptons.size(); ++it) {
for (unsigned int jt = 0; jt < it; ++jt) {
    double candidatemass = (
        leptons.at(it).mom() + leptons.at(jt).mom()
    ).mass();
```

```
        if (leptons.at(it).pid() == -leptons.at(jt).pid()
            && abs(candidatemass - 91.1876*GeV) < massdiff
        ) {
        bosoncandidateexists = true;
        massdiff = abs(candidatemass - 91.1876*GeV);
        if (leptons.at(it).pid() > 0) {
            muon = leptons.at(it);
            antimuon = leptons.at(jt);
        }
        else {
            muon = leptons.at(jt);
            antimuon = leptons.at(it);
        }
        }
        else continue;
    }
    }

    if (!(bosoncandidateexists)) vetoEvent;
    MSG_DEBUG("Found Z-boson candidate with mass "
        << (muon.mom() + antimuon.mom()).mass()/GeV
        << "GeV");

    // Fill histograms with selected events
    const double rap_Z = (
        muon.mom() + antimuon.mom()
    ).rap();
    const double pT_Z = (
        muon.mom() + antimuon.mom()
    ).pT()/GeV;
    if (pT_Z <= _minptZ) vetoEvent;
    MSG_DEBUG("\tZ-boson pt: " << pT_Z);
    MSG_DEBUG("\tZ-boson y: " << rap_Z);

    const double thetastar = acos(
        tanh((antimuon.mom().eta() - muon.mom().eta())/2)
    );

    /// Fill signal histograms
    vector<double> binedges_Ystar = {
        0.5, 1.0, 1.5, 2.0, 2.5
    };
    vector<double> binedges_Yboost = {
        0.5, 1.0, 1.5, 2.0, 2.5
```

```
    };

    for (auto jets: _jetcollections) {
    // Fill jet related histograms
    _h["NJets"+jets.first] -> fill(jets.second.size());

    double rap_Jet1 = jets.second.at(0).rap();

    double rap_star = 0.5 * abs(rap_Z - rap_Jet1);
    double rap_boost = 0.5 * abs(rap_Z + rap_Jet1);

    for(auto _ystar: binedges_Ystar){
        for(auto _yboost: binedges_Yboost){
        if(_ystar + _yboost > 3.) continue;
        if((rap_star < _ystar) && (rap_boost < _yboost)){

            // The histograms are named with
            // the left bin border
            double _ystar_label = _ystar - 0.5;
            double _yboost_label = _yboost - 0.5;

            MSG_DEBUG(
                "Selected y*-yb bin: Ys" << _ystar_label
                << "Yb"<< _yboost_label
            );
            string _hist_ZPt_ident = "ZPt"+jets.first
                +"Ys"+to_string(_ystar_label)+"Yb"
                +to_string(_yboost_label);

            // Fill the histograms
            _h[_hist_ZPt_ident]->fill(pT_Z);

            // End the loop,
            // when a matching bin has been found
            goto theEnd;
        }
        else continue;
        }
    }
    theEnd:;
    }
}
```

```
    /// Normalise histograms etc., after the run
    void finalize() {
        /// Normalise histograms
        const double sf = crossSection()
                          /picobarn/sumOfWeights();

        for(auto const& _hist : _h){
            scale(_hist.second, sf);
        }
    }
    ///@}

    /// @name Histograms
    ///@{
    map<string,Histo1DPtr> _h;
    ///@}

    /// @name Selections
    ///@{
    // mass window around Z-boson PDG mass
    const double _massdiff = 20*GeV;
    const double _minptZ = 25*GeV;
    // minimum pT of hardest jet
    const double _minjet1pt = 20*GeV;
    // minimum jet pT
    const double _jetpt = 10*GeV;
    // maximum absolute jet y
    const double _maxabsjetrap = 2.4;
    // maximium number of leptons
    const size_t _maxnleptons = numeric_limits<size_t>::max();
    // maximum absolute lepton eta
    const double _maxleptoneta = 2.4;
    // minimum lepton pT
    const double _minleptonpt = 25*GeV;
    // DeltaR between leptons and jets
    // to clean former from latter
    const double _lepCleaningDeltaR = 0.3;
    ///@}

    };

    DECLARE_RIVET_PLUGIN(ZplusJet_3);
}
```

A.1.3 NP- MPI- & Hadronization-Corrections

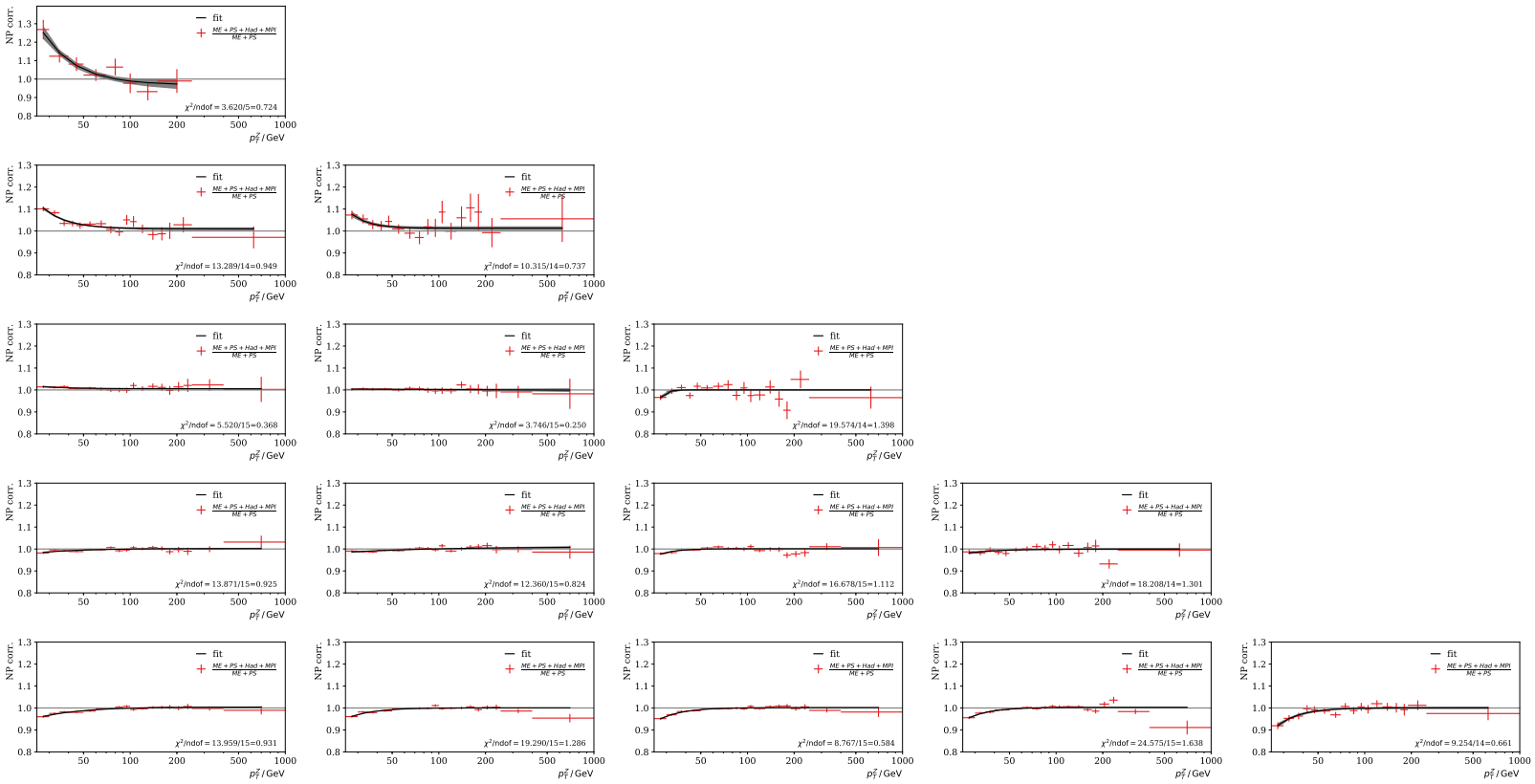


Figure A.1: Non-perturbative correction factors derived from Herwig generations using the full generation chain (ME+PS+Had+MPI) and the partial generation chain (ME+PS) at LO accuracy in QCD are shown. A fitted parametric function smooths statistical fluctuations. The goodness-of-fit is estimated using the value of the objective function eq. (5.15), denoted as χ^2, divided by the number of degrees of freedom n_{dof}.

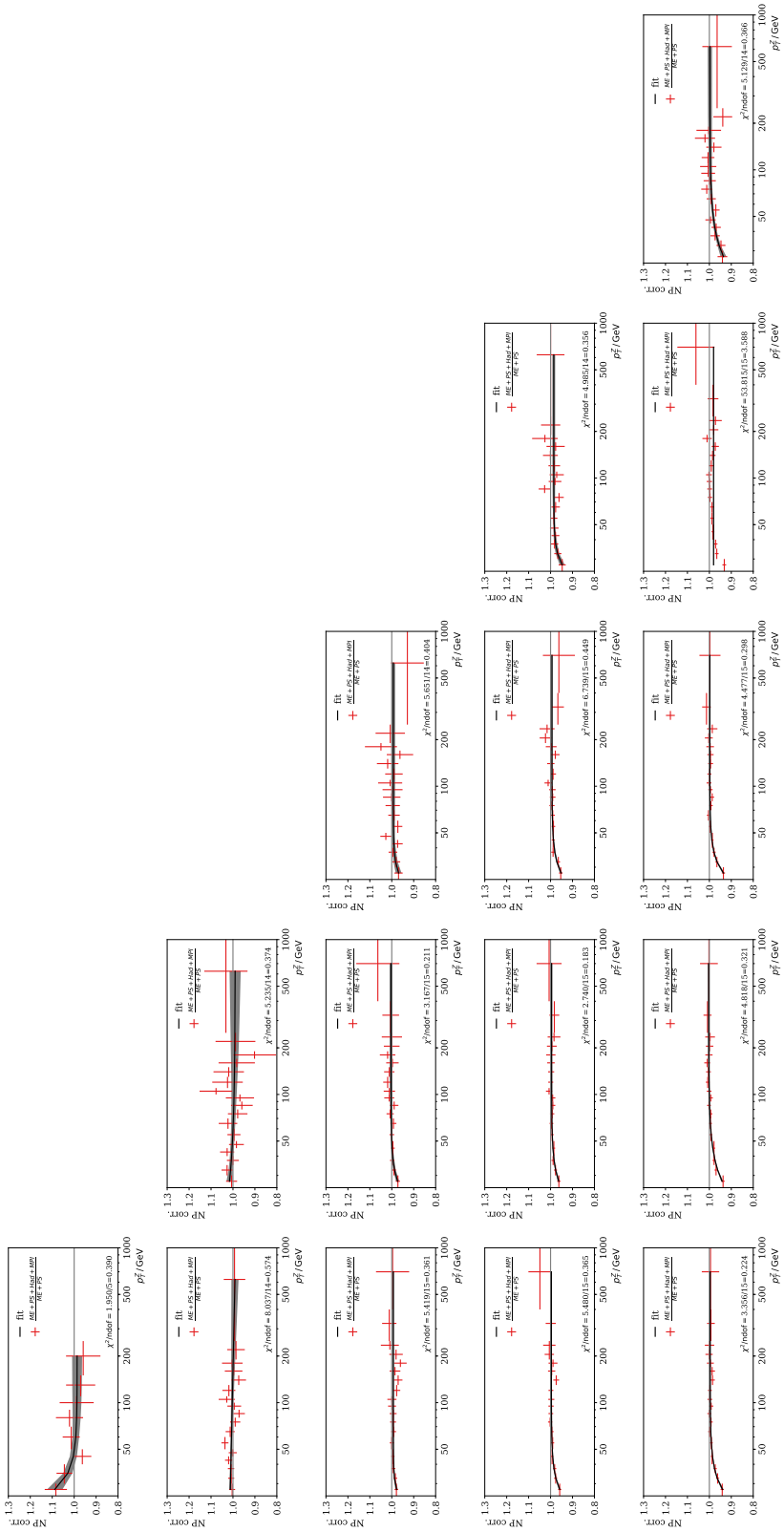


Figure A.2: Non-perturbative correction factors derived from Herwig generations using the full generation chain (ME+PS+Had+MPI) and the partial generation chain (ME+PS) at NLO accuracy in QCD are shown. The correction factors are in general smaller than the ones obtained at LO accuracy shown in fig. A.1.

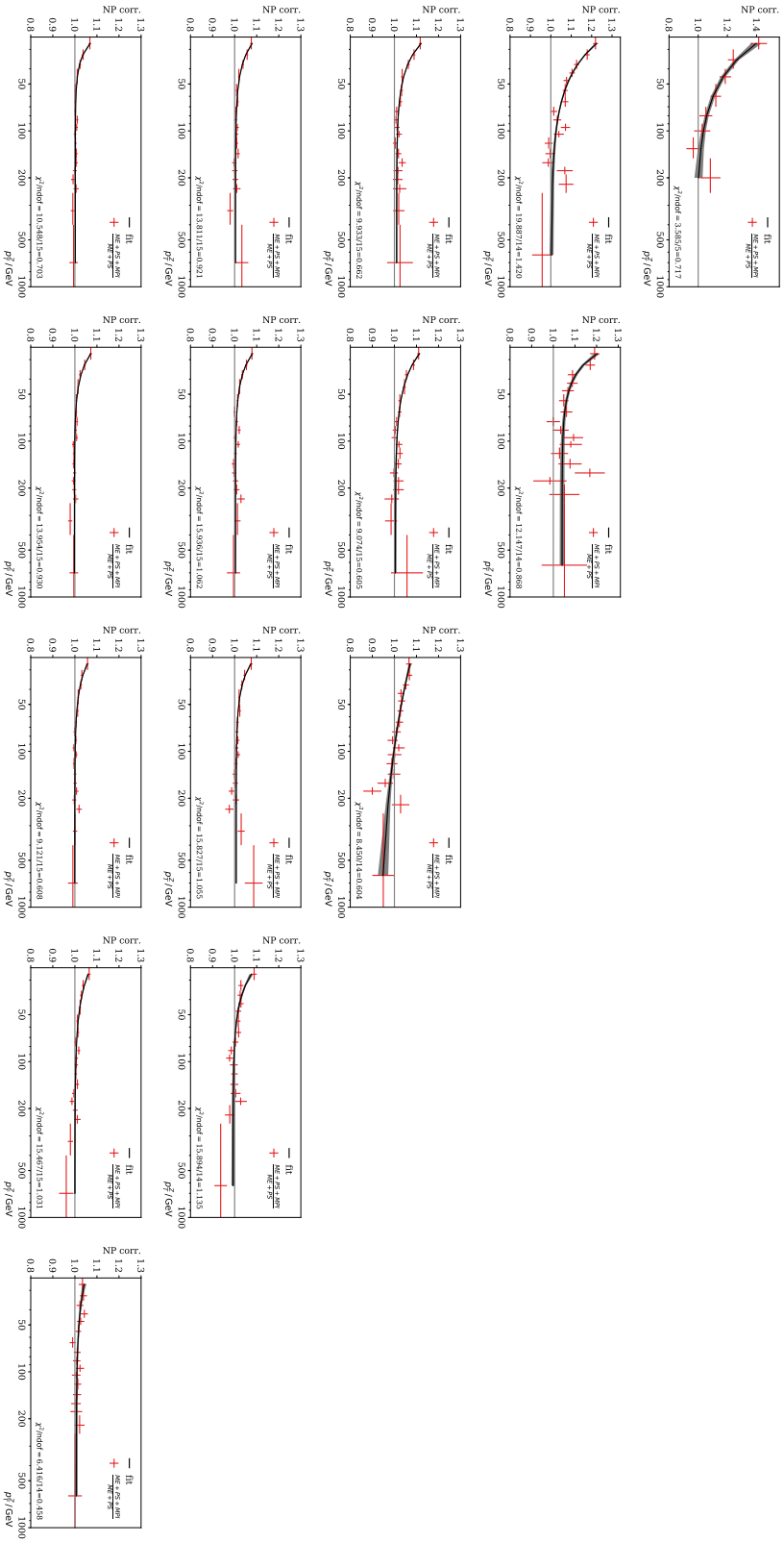


Figure A.3: MPI correction factors derived from Herwig generations using the partial generation chain ME+PS+MPI in the nominator and the partial generation chain ME+PS in the denominator at LO accuracy in QCD are shown. A fitted parametric function smooths statistical fluctuations. The goodness-of-fit is estimated using the value of the objective function eq. (5.15), denoted as χ^2, divided by the number of degrees of freedom n_{dof}.

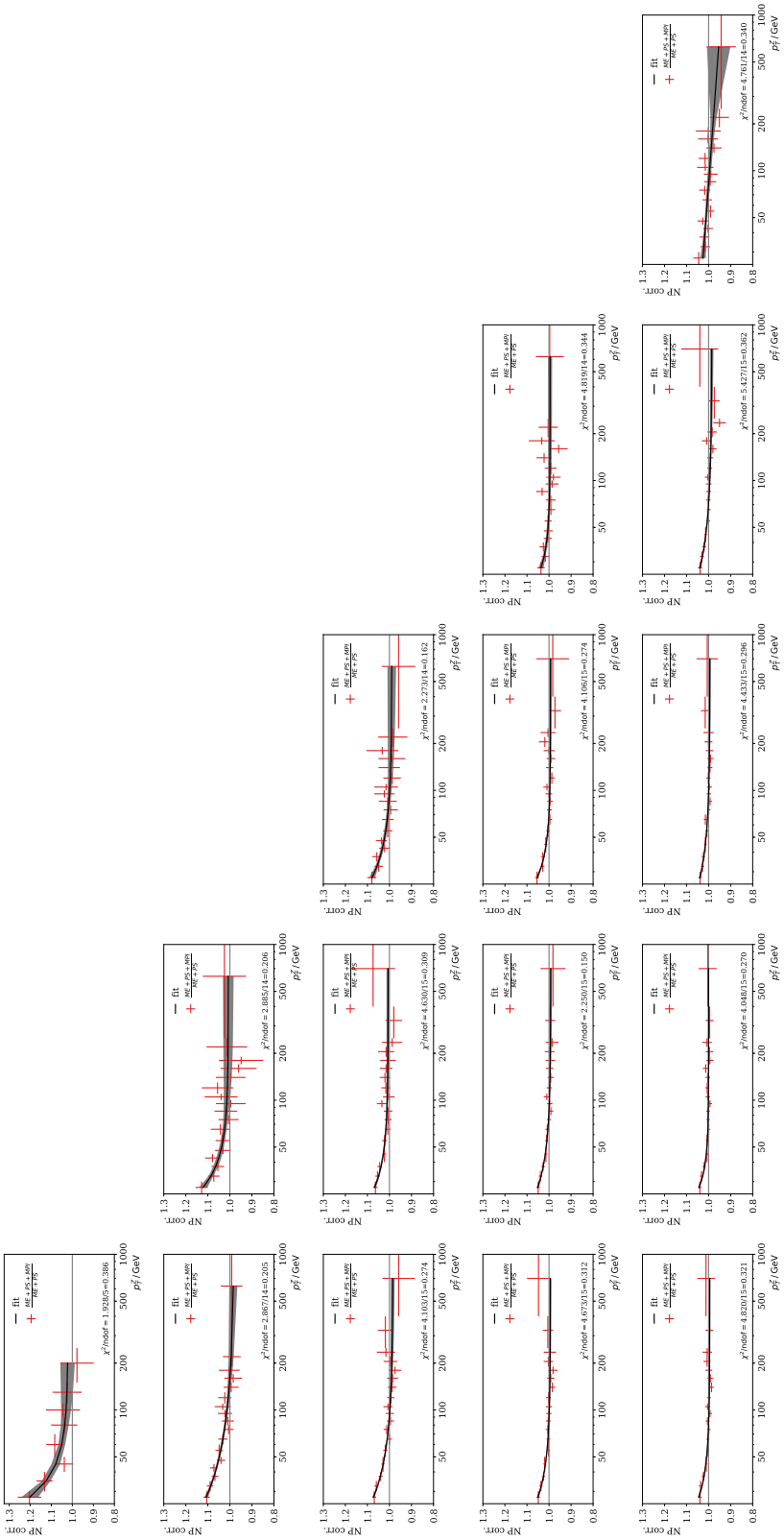


Figure A.4: MPI correction factors derived from Herwig generations using the partial generation chain ME+PS+MPI in the nominator and the partial generation chain ME+PS in the denominator at NLO accuracy in QCD are shown. The correction factors are in general smaller than the ones obtained at LO accuracy shown in fig. A.3.

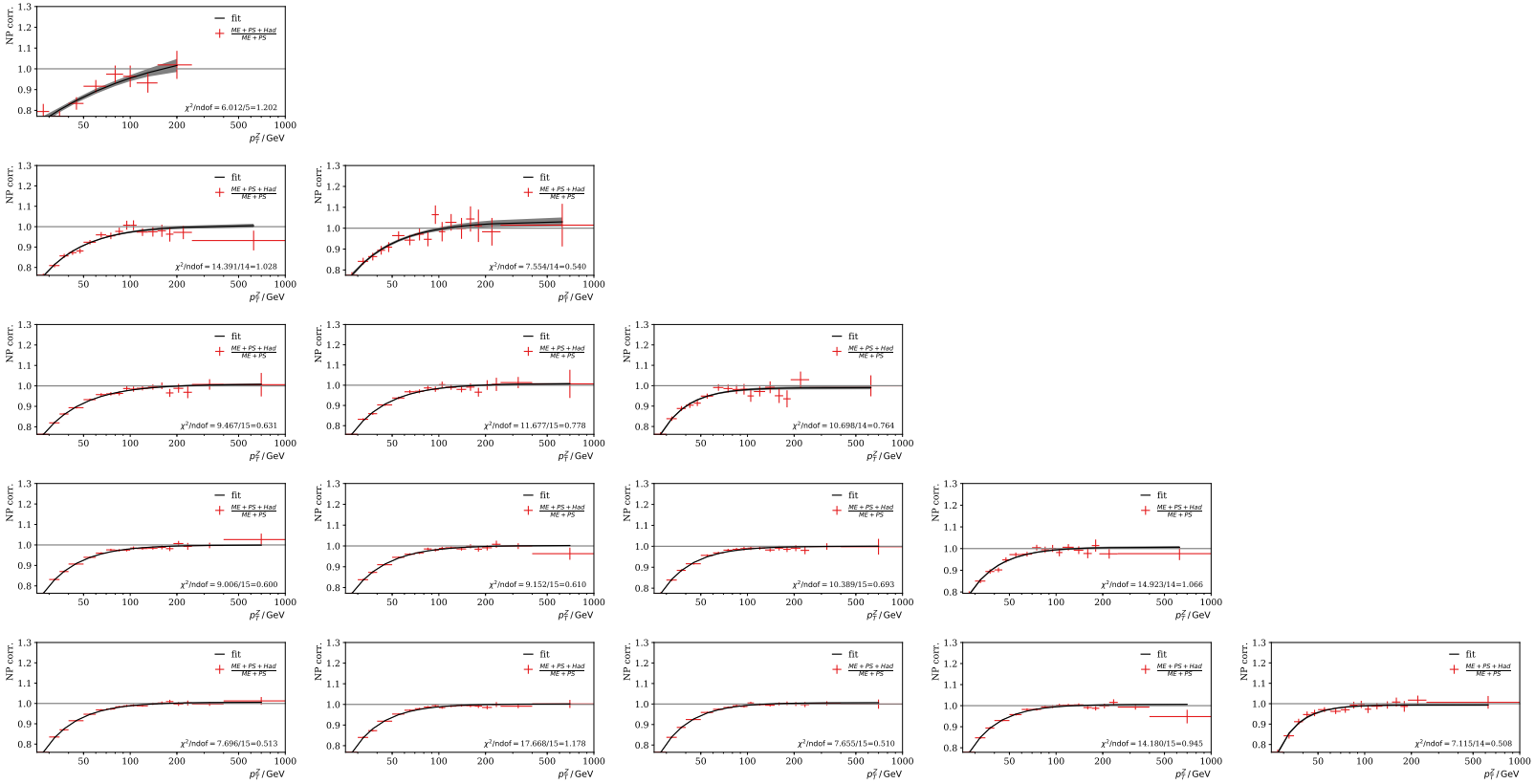


Figure A.5: Hadronization correction factors derived from Herwig generations using the partial generation chain ME+PS+Had in the nominator and the partial generation chain ME+PS in the denominator at LO accuracy in QCD are shown. A fitted parametric function smooths statistical fluctuations. The goodness-of-fit is estimated using the value of the objective function eq. (5.15), denoted as χ^2, divided by the number of degrees of freedom n_{dof}.

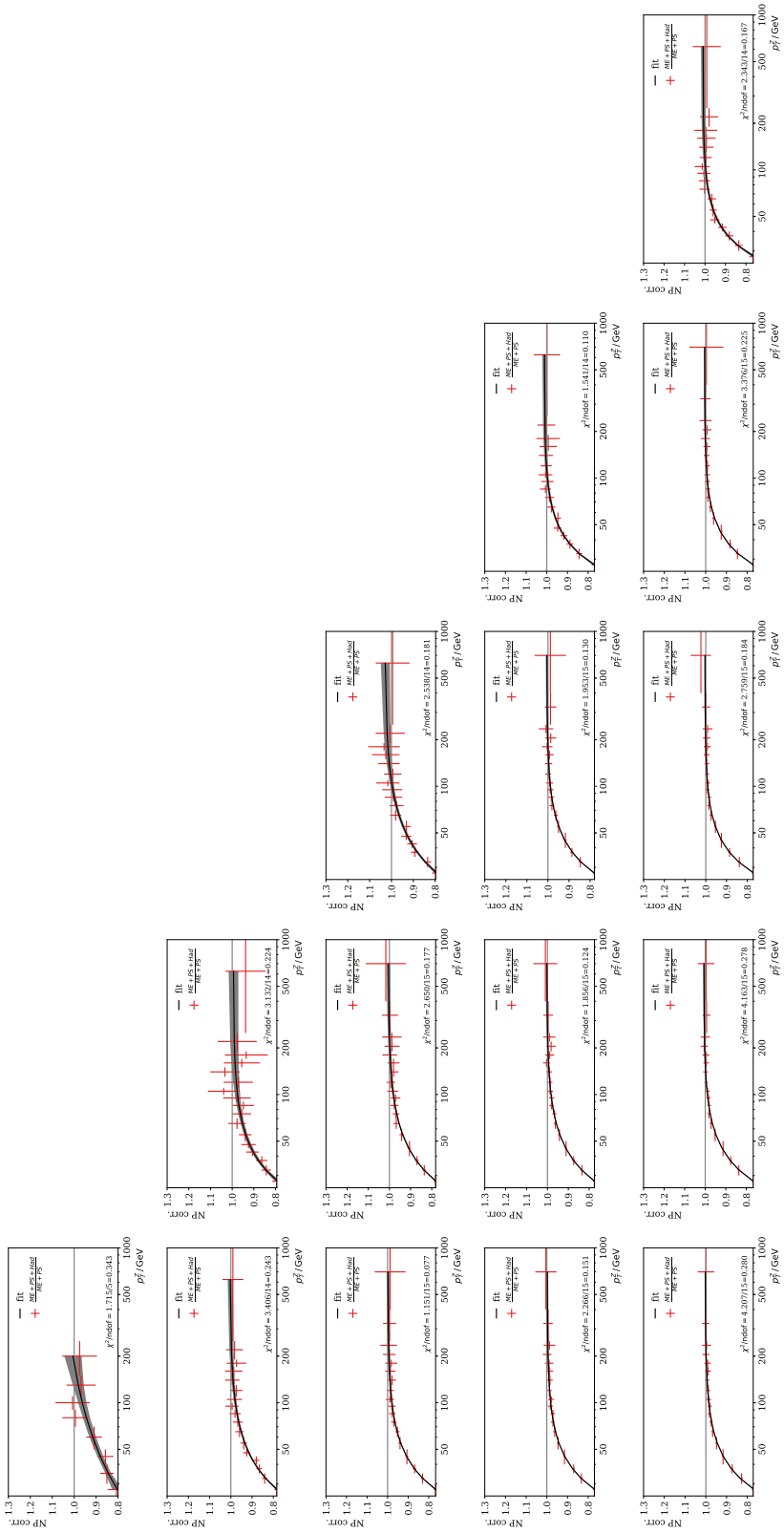


Figure A.6: Hadronization correction factors derived from Herwig generations using the partial generation chain ME+PS+Had in the nominator and the partial generation chain ME+PS in the denominator at NLO accuracy in QCD are shown. No significant difference between the correction factors at NLO and LO accuracy, shown in fig. A.5, are observed.

A.2 Comparisons of Data with Simulated Data

Comparisons of the measured data (see section 5.2) with simulations for background and signal processes (see section 5.3) are shown. After application of all corrections, quality criteria and selections (see section 5.1) the resulting histograms for observables on the muons, the dimuon system and the hardest jet are compared for the four data taking periods inclusive in y_b-y^*. After that the same observables are compared for each y_b-y^*-bin using the combined dataset stacking the yields of the individual data-taking periods. The shown uncertainties include the full treatment of statistical and systematic effects as described in sections 5.5 and 5.6.2.

A.2.1 Muon Observables

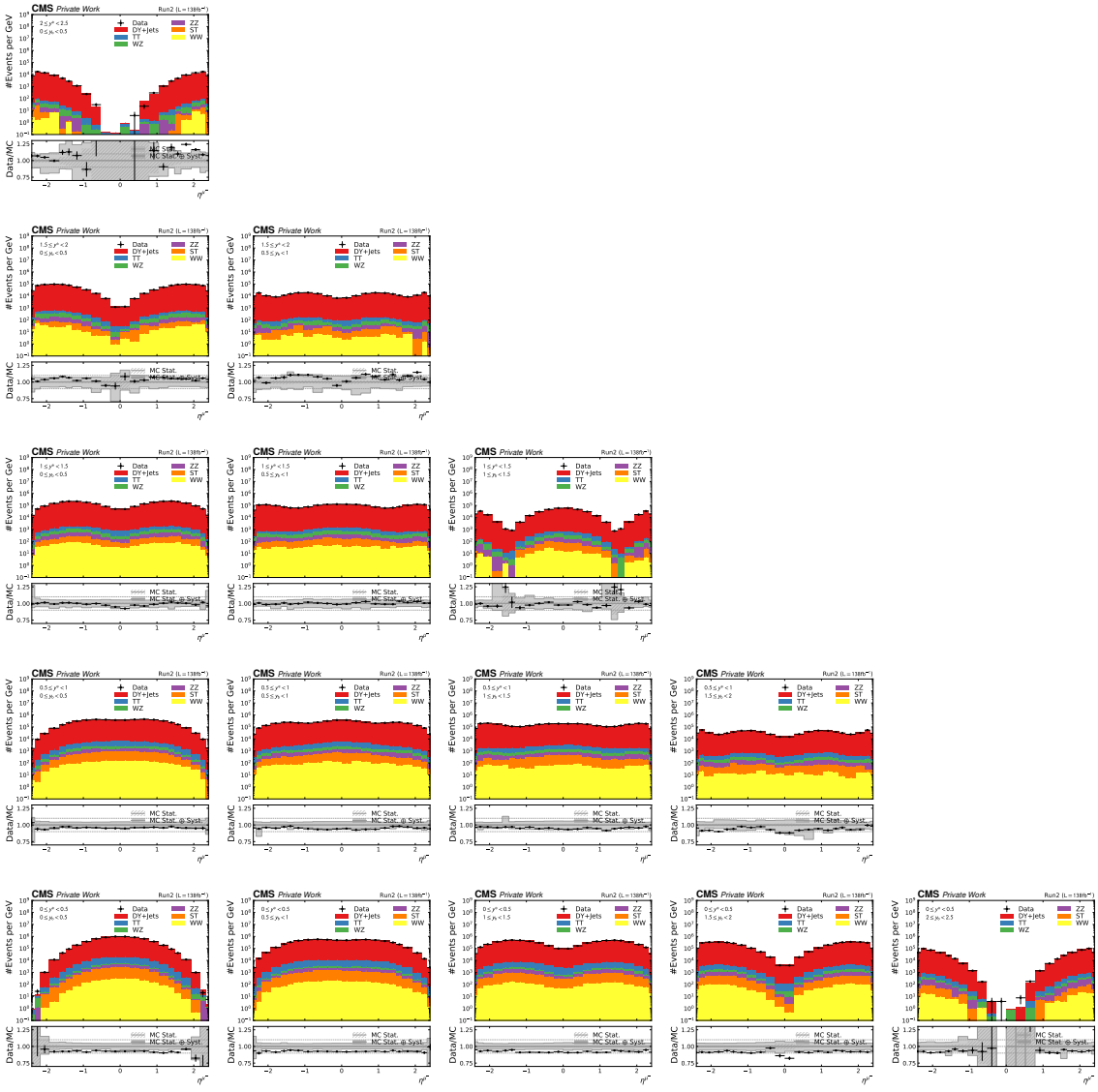


Figure A.7: Comparison of the pseudorapidity of the positively charged muon selected for the dimuon system reconstruction η^{μ^-} for each y_b-y^*-bin and the combined dataset.

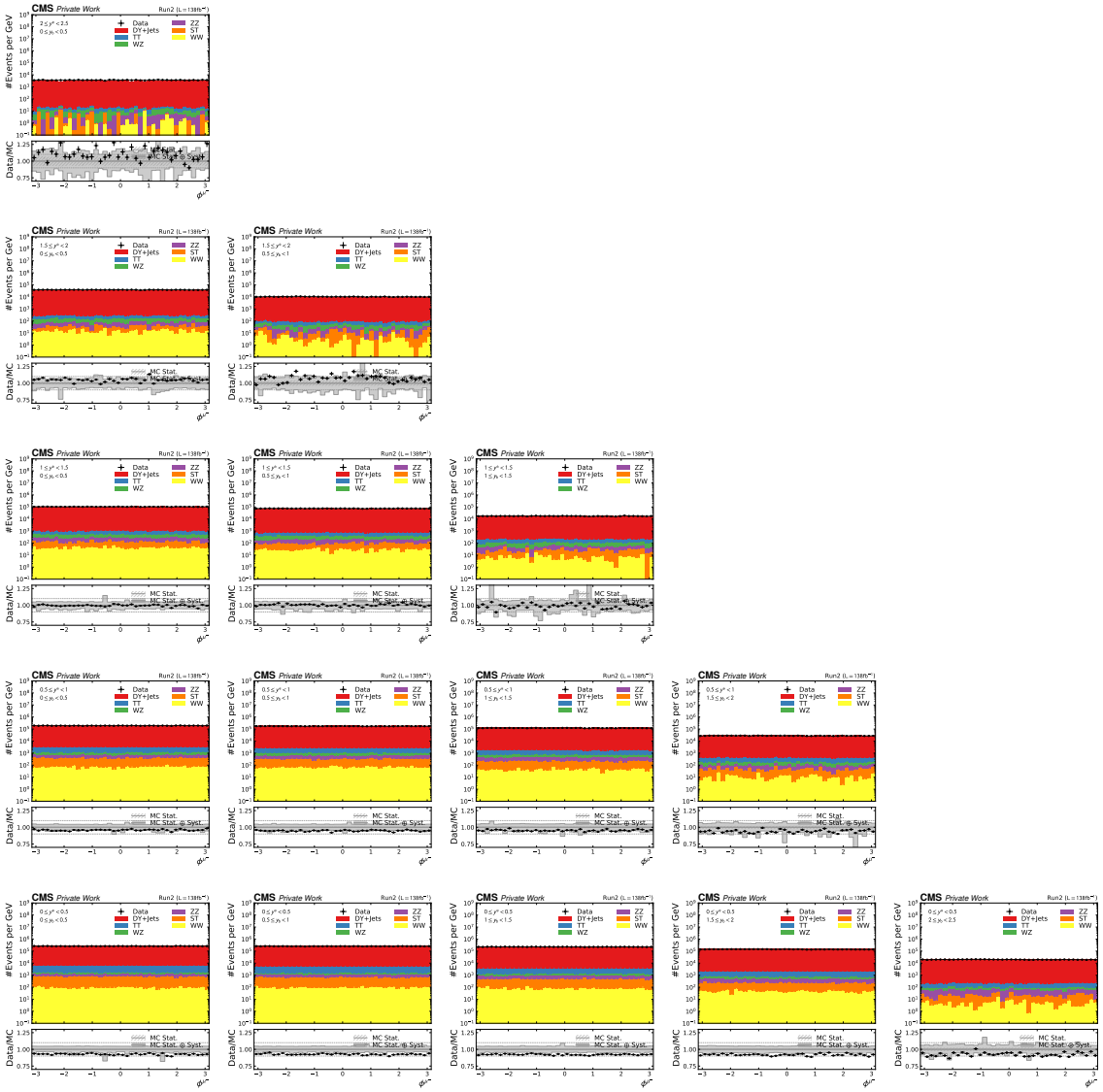


Figure A.8: Comparison of the azimuth angle of the positively charged muon selected for the dimuon system reconstruction ϕ^{μ^-} for each y_b-y^*-bin and the combined dataset.

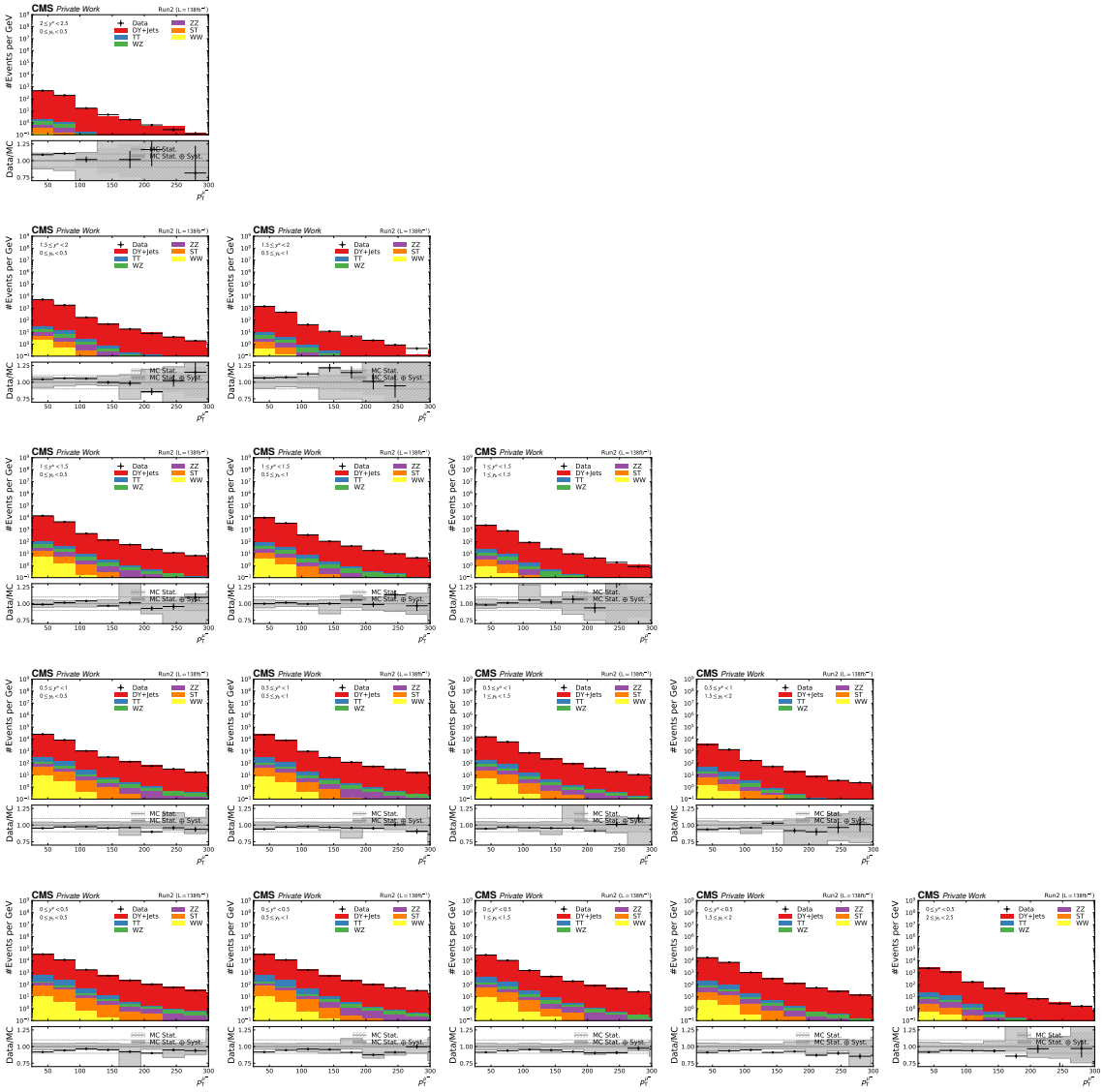


Figure A.9: Comparison of the transverse momentum of the positively charged muon selected for the dimuon system reconstruction $p_T^{\mu^-}$ for each y_b-y^*-bin and the combined dataset.

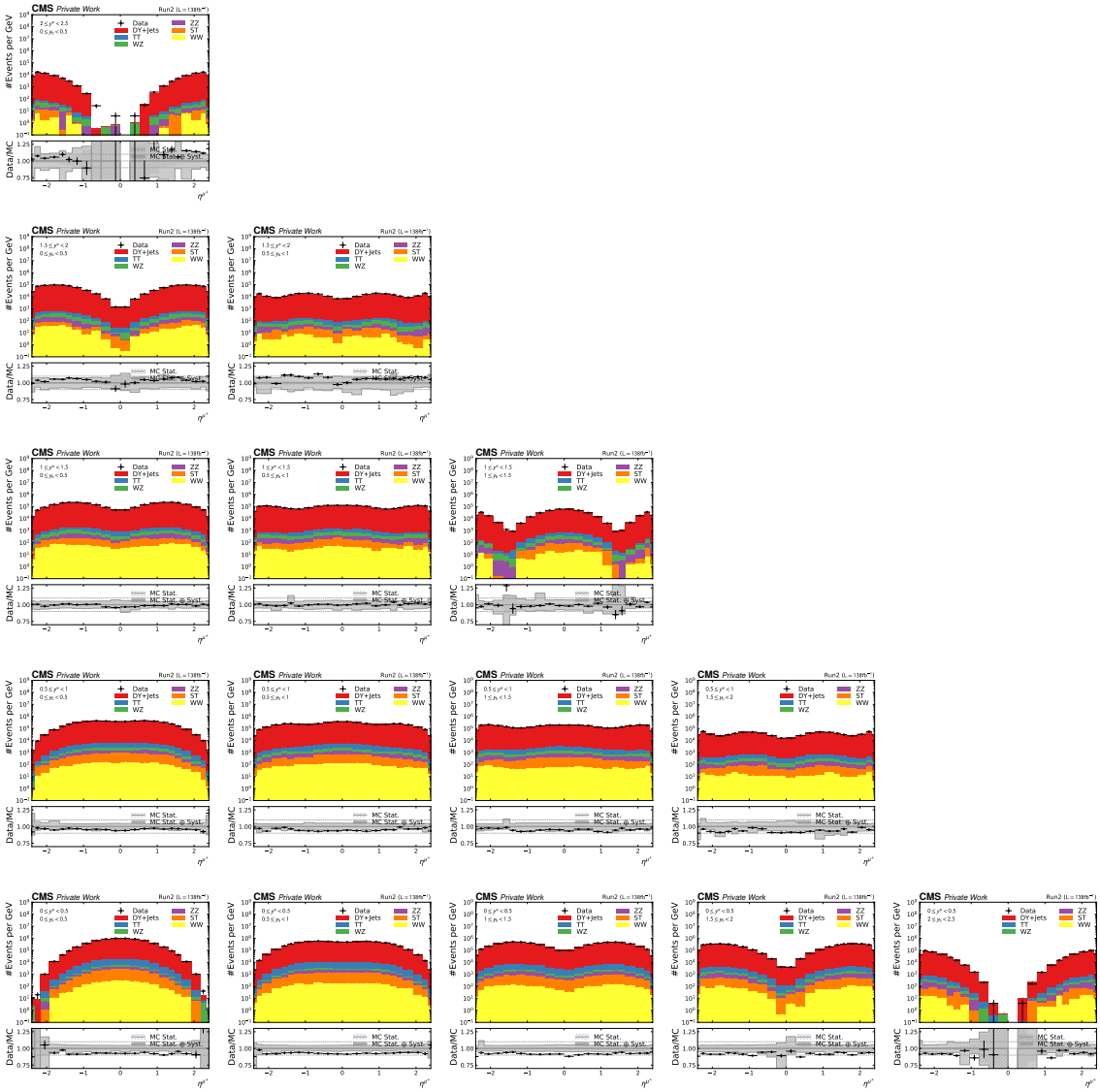


Figure A.10: Comparison of the pseudorapidity of the positively charged muon selected for the dimuon system reconstruction η^{μ^-} for each y_b-y^*-bin and the combined dataset.

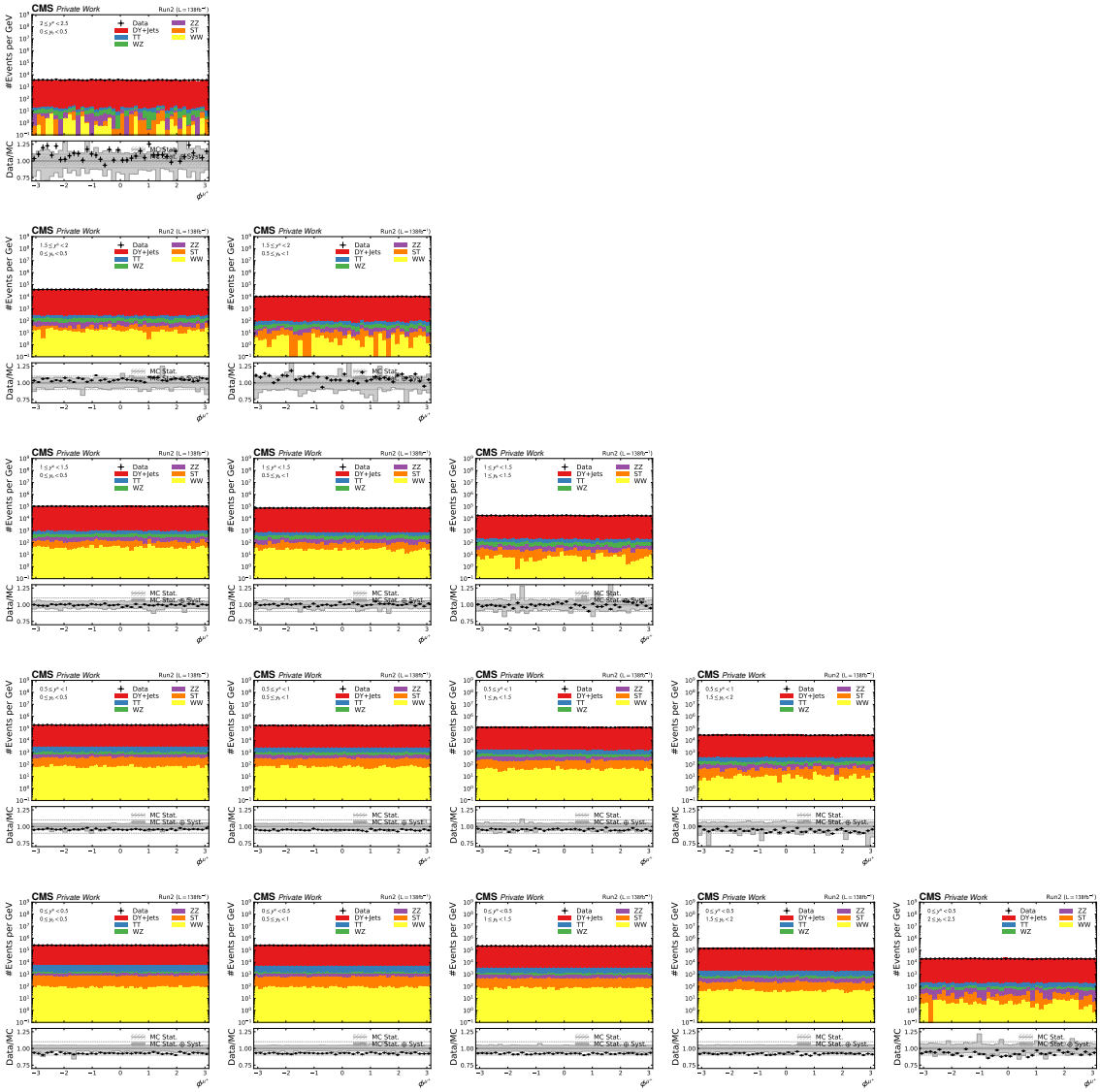


Figure A.11: Comparison of the azimuth angle of the positively charged muon selected for the dimuon system reconstruction ϕ^{μ^-} for each y_b-y^*-bin and the combined dataset.

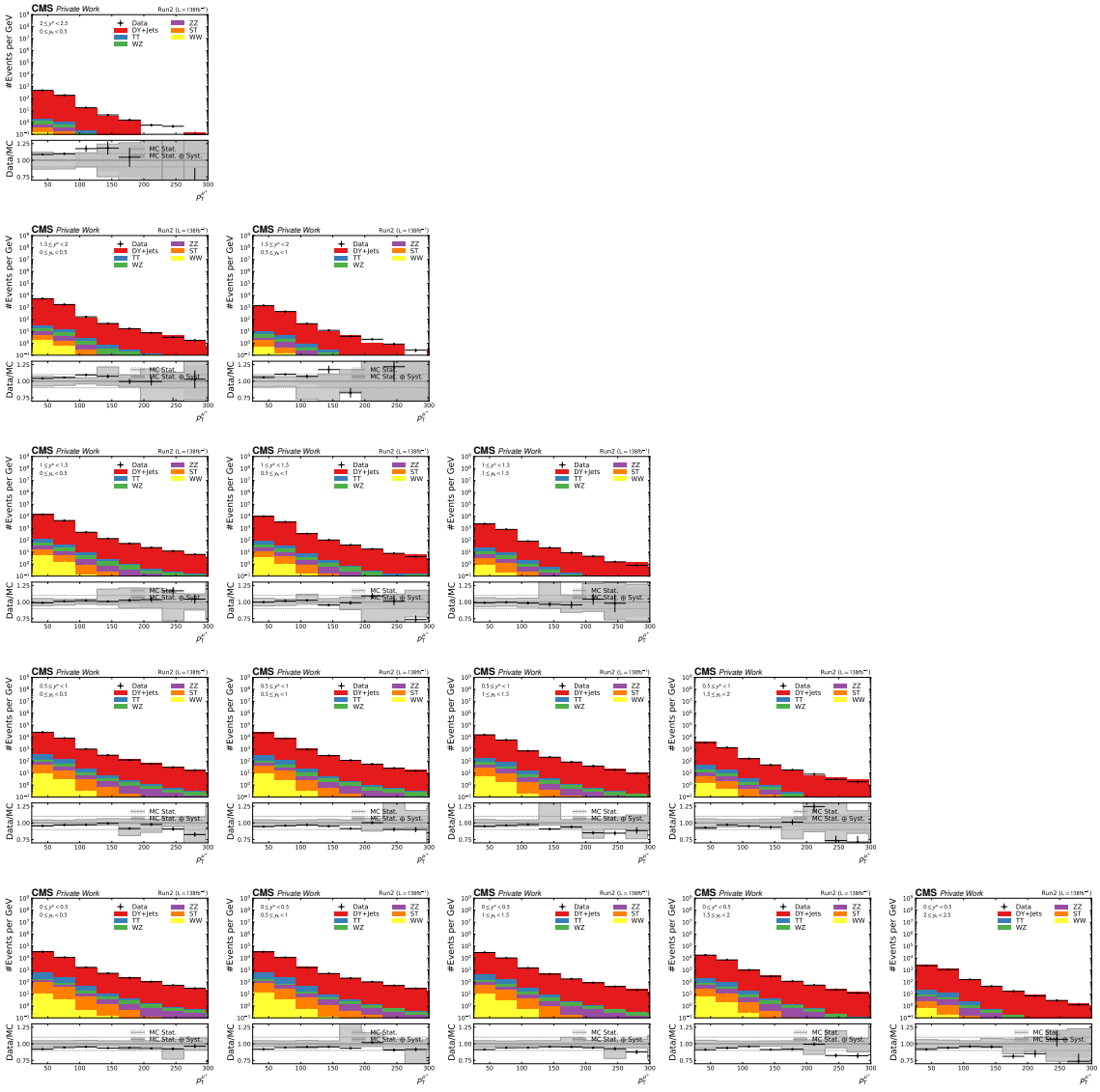


Figure A.12: Comparison of the transverse momentum of the positively charged muon selected for the dimuon system reconstruction $p_T^{\mu^-}$ for each y_b-y^*-bin and the combined dataset.

A.2.2 Observables on the Dimuon System

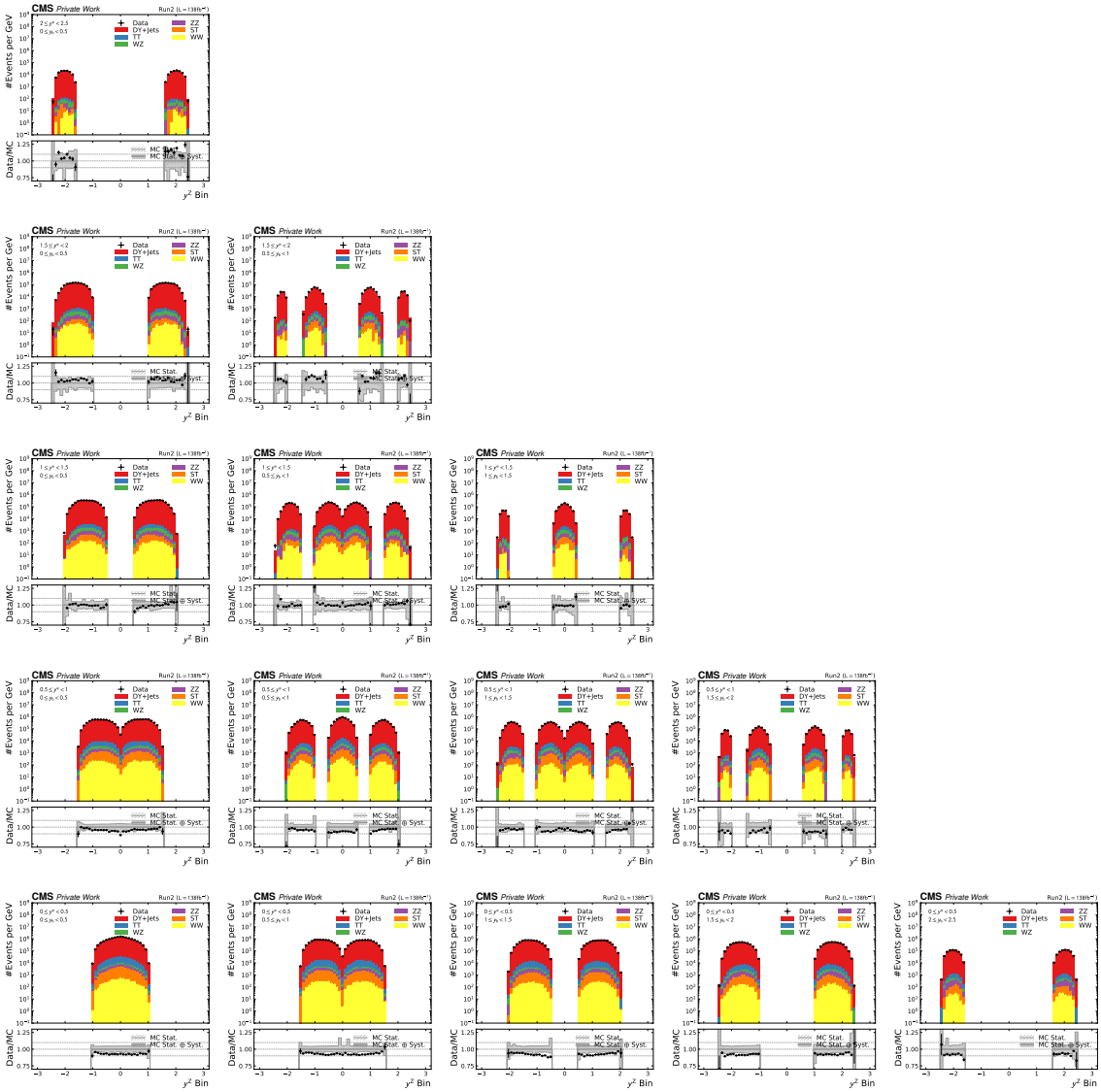


Figure A.13: Comparison of the rapidity of the dimuon system y^Z for each y_b-y^*-bin and the combined dataset.

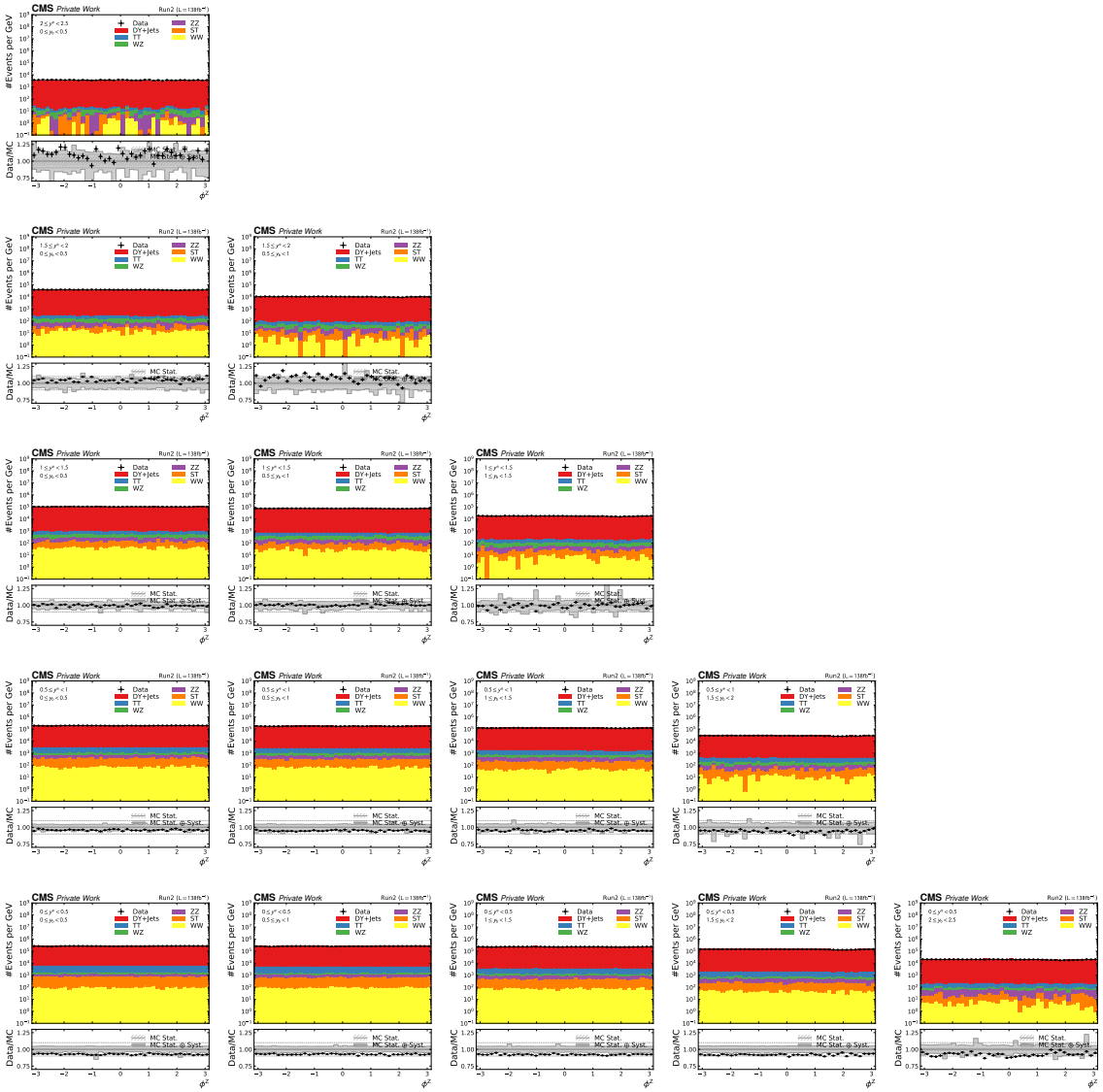


Figure A.14: Comparison of the azimuth angle of the dimuon system ϕ^Z for each y_b-y^*-bin and the combined dataset.

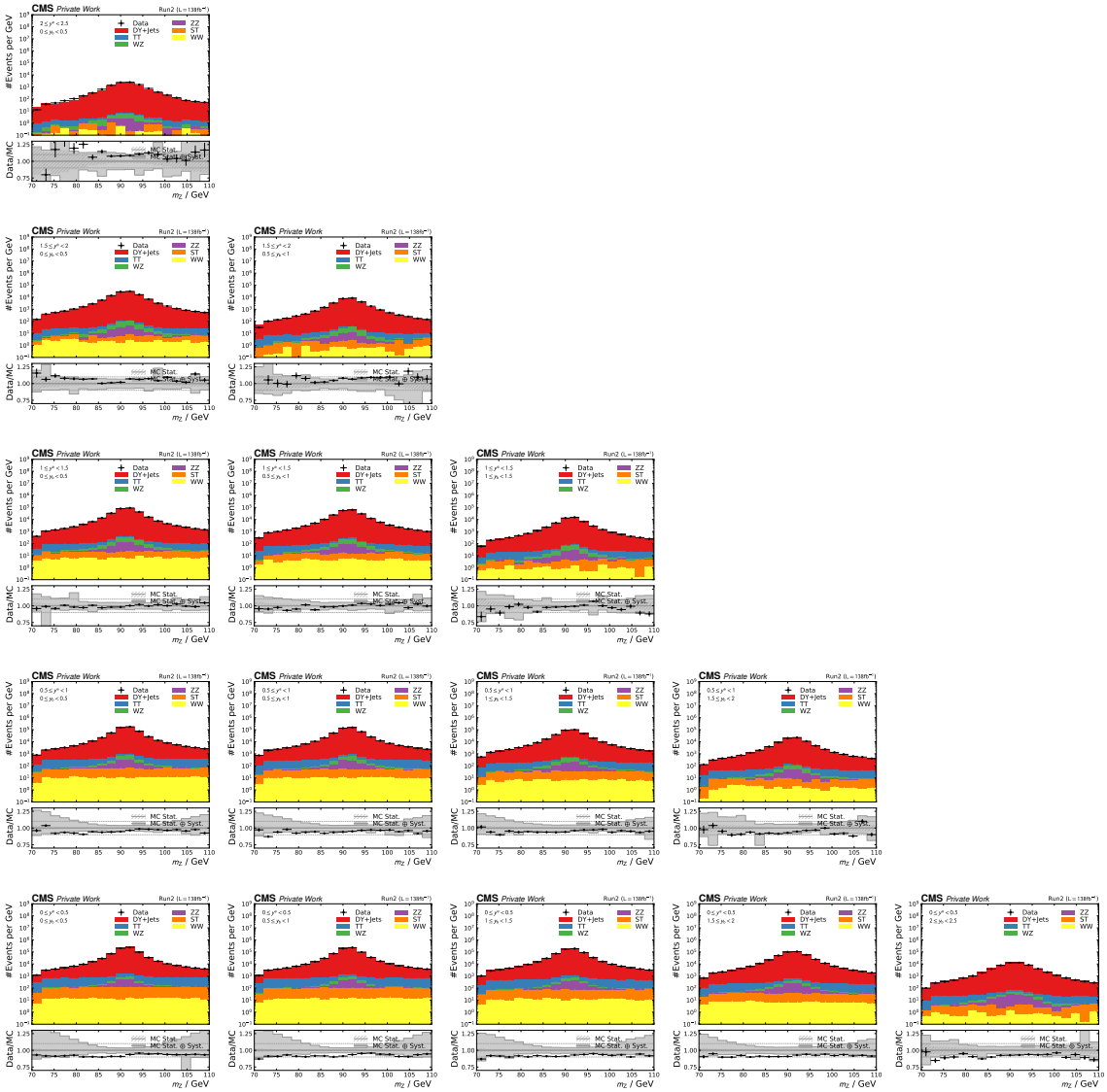


Figure A.15: Comparison of the invariant mass of the dimuon system m_Z for each y_b-y^*-bin and the combined dataset.

A.2.3 Jet Observables

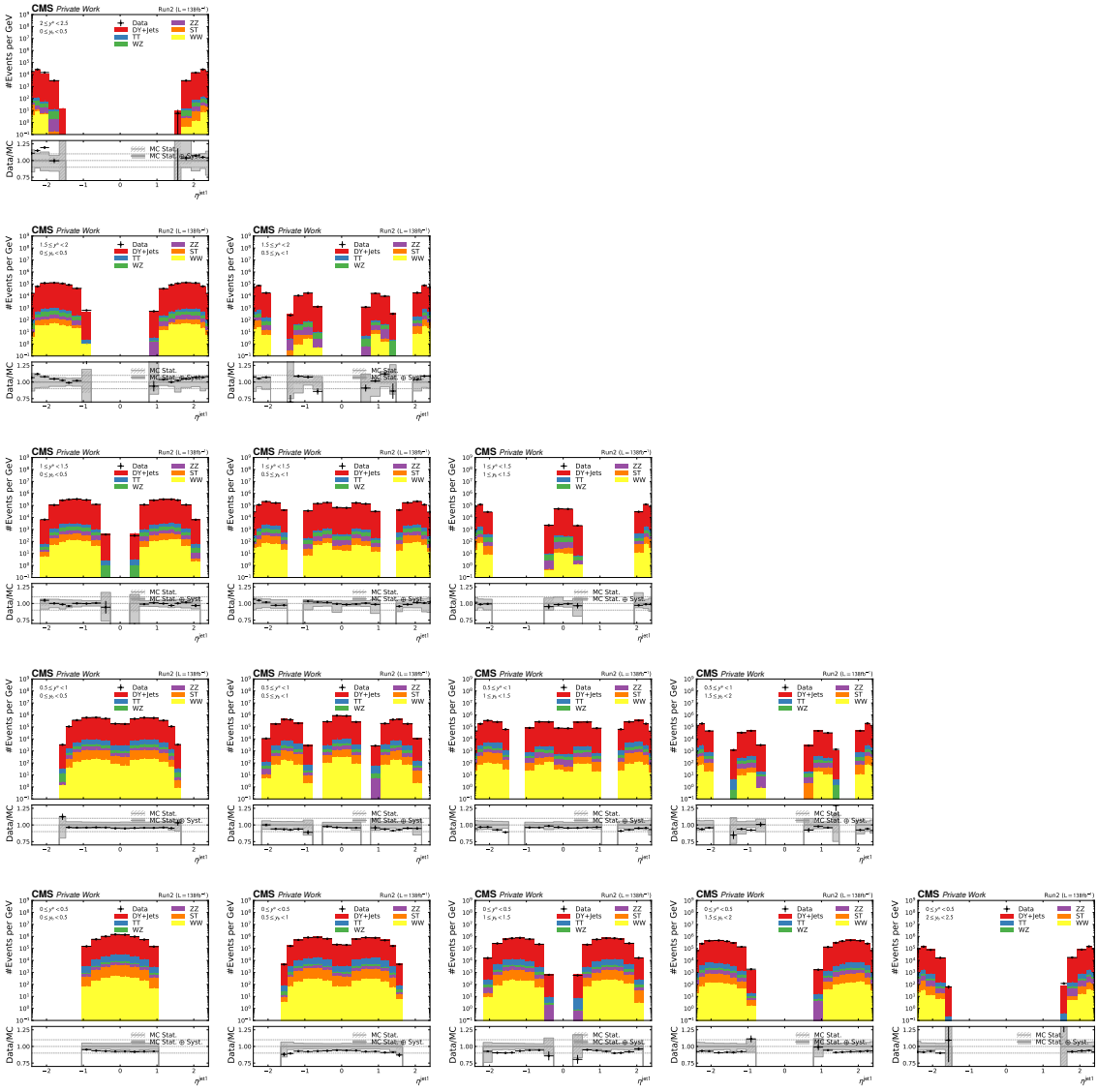


Figure A.16: Comparison of the pseudorapidity of the hardest jet η^{jet1} for each y_b-y^*-bin and the combined dataset.

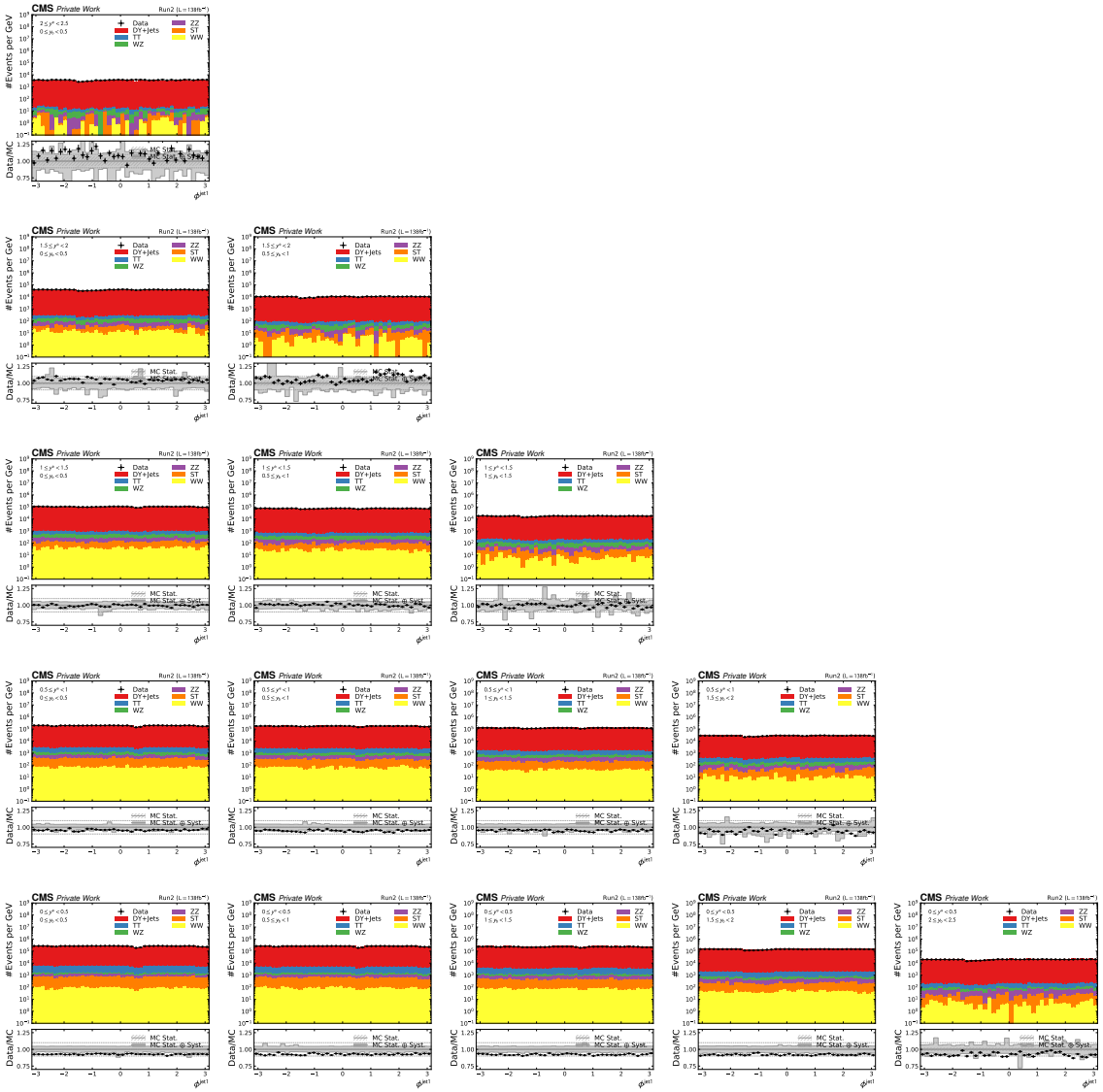


Figure A.17: Comparison of the azimuth angle of the hardest jet ϕ^{jet1} for each y_b-y^*-bin and the combined dataset.

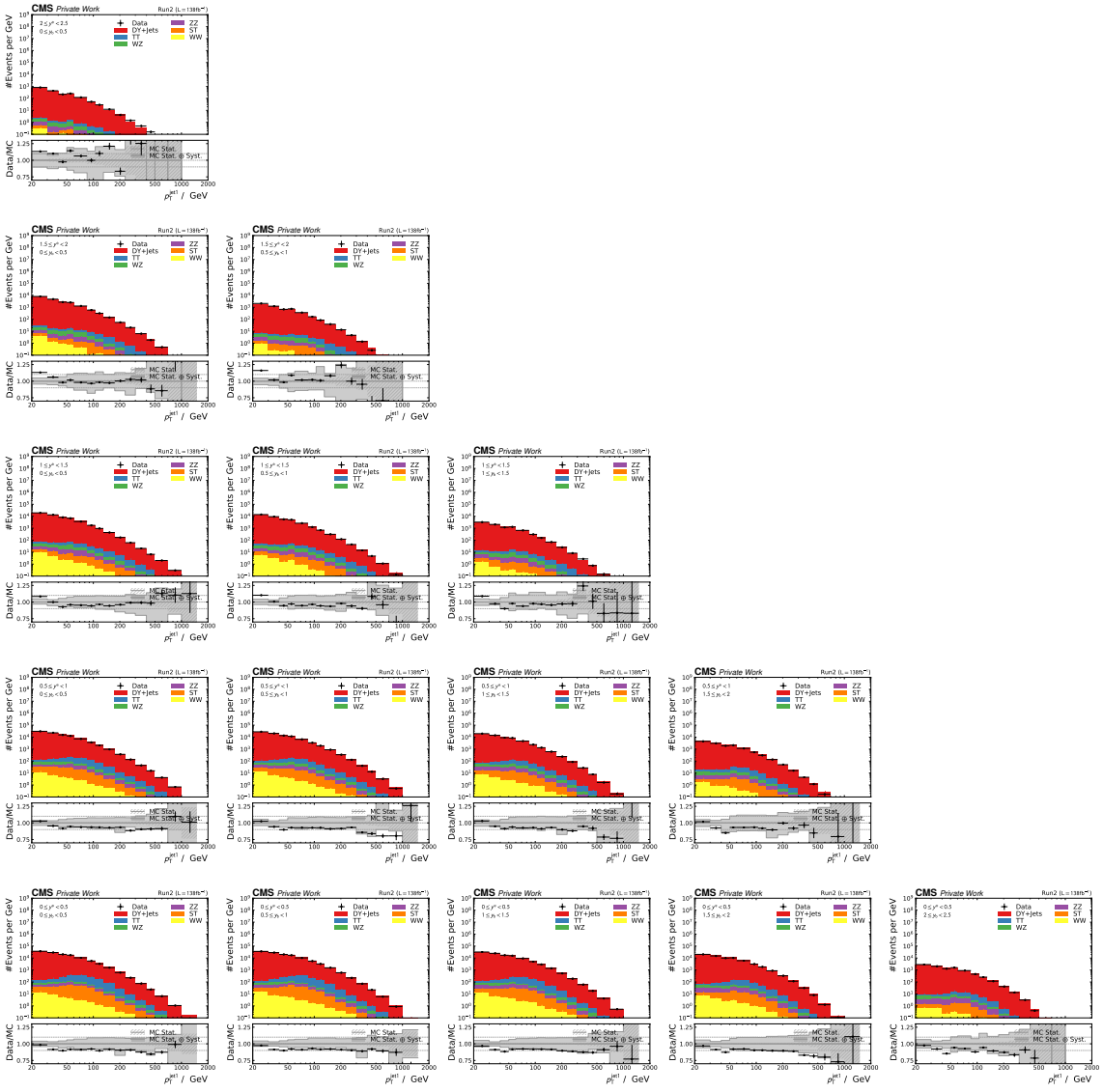


Figure A.18: Comparison of the transverse momentum of the hardest jet p_T^{jet1} for each y_b-y^*-bin and the combined dataset.

A.2.4 Unfolding Input Yields

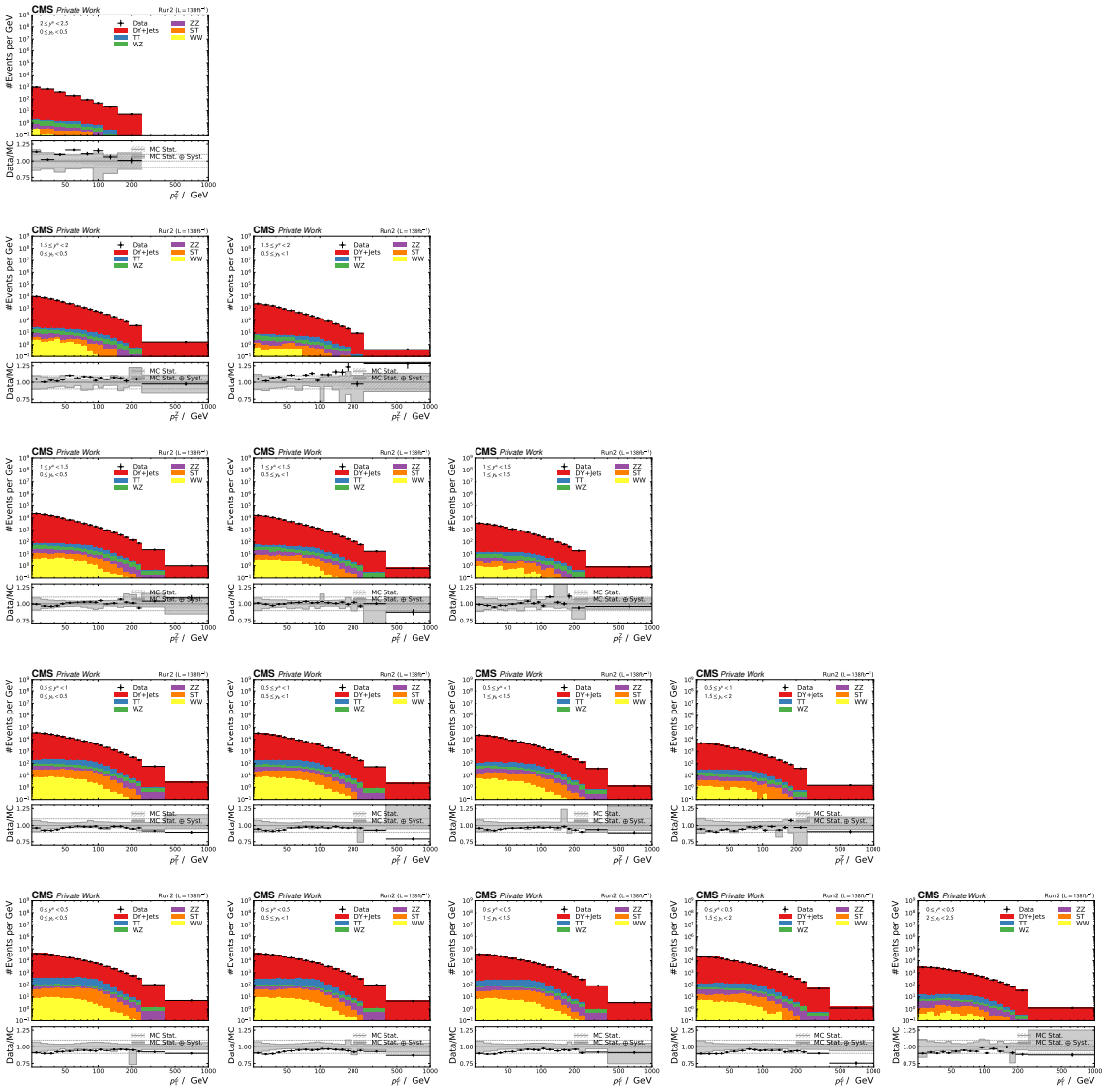


Figure A.19: Comparison of the transverse momentum of the dimuon system p_T^Z for each y_b-y^*-bin and the combined dataset. The observed differential shapes of the event yields predicted by the stacked signal and background simulations agree with the ones selected in data within uncertainties. A systematic bias for an inclusive normalization factor is indicated by a shift of the simulation. The normalization factor grows from approximately 95% to 110% with increasing y^*. No dependence on y_b is observed.

A.3 Unfolding

A.3.1 Acceptances and Fakerates in All Bins

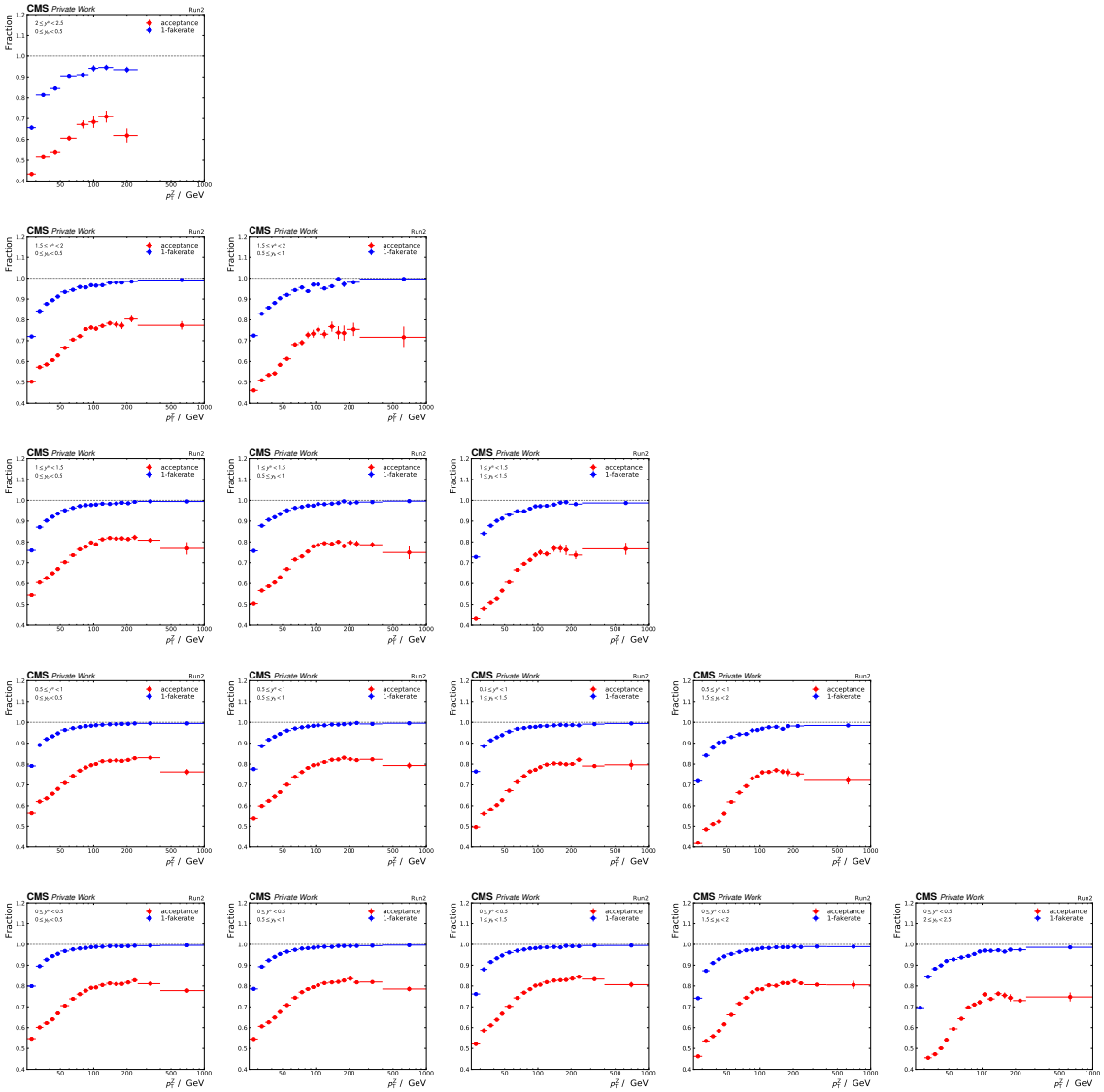


Figure A.20: Full set of derived acceptances and fakerates (shown as 1-fakerate) (see section 5.5.2) constructed for the unfolding of the full Run 2 data. The fakerate is maximal at low p_T^Z and converges towards 0 for high p_T^Z. The acceptance is minimal at low p_T^Z and reaches a plateau for high p_T^Z. Towards the boundaries of the analysed phase space the acceptances drop again and the convergence of the fakerates is slowed.

A.3.2 Cross-Checks of Unfolding

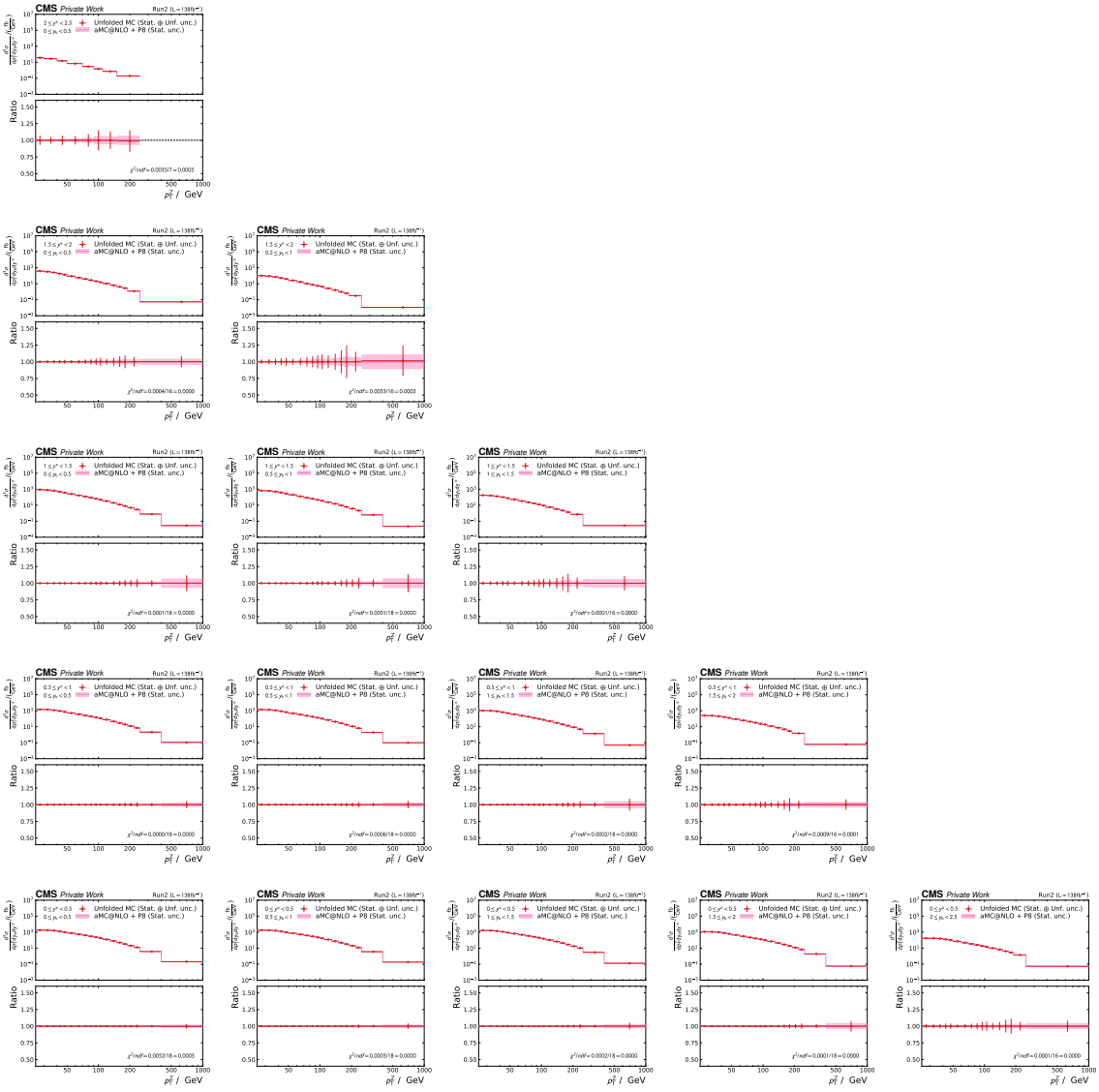


Figure A.21: Closure of the unfolding procedure used for the combined Run 2 data. The migration matrix constructed from the combined set of simulated events is used for performing the unfolding on the simulated signal yields on reconstruction level. The unfolded results (red points) with statistical uncertainties are compared to the corresponding predictions at generation level (pink band). The two sets are in perfect agreement.

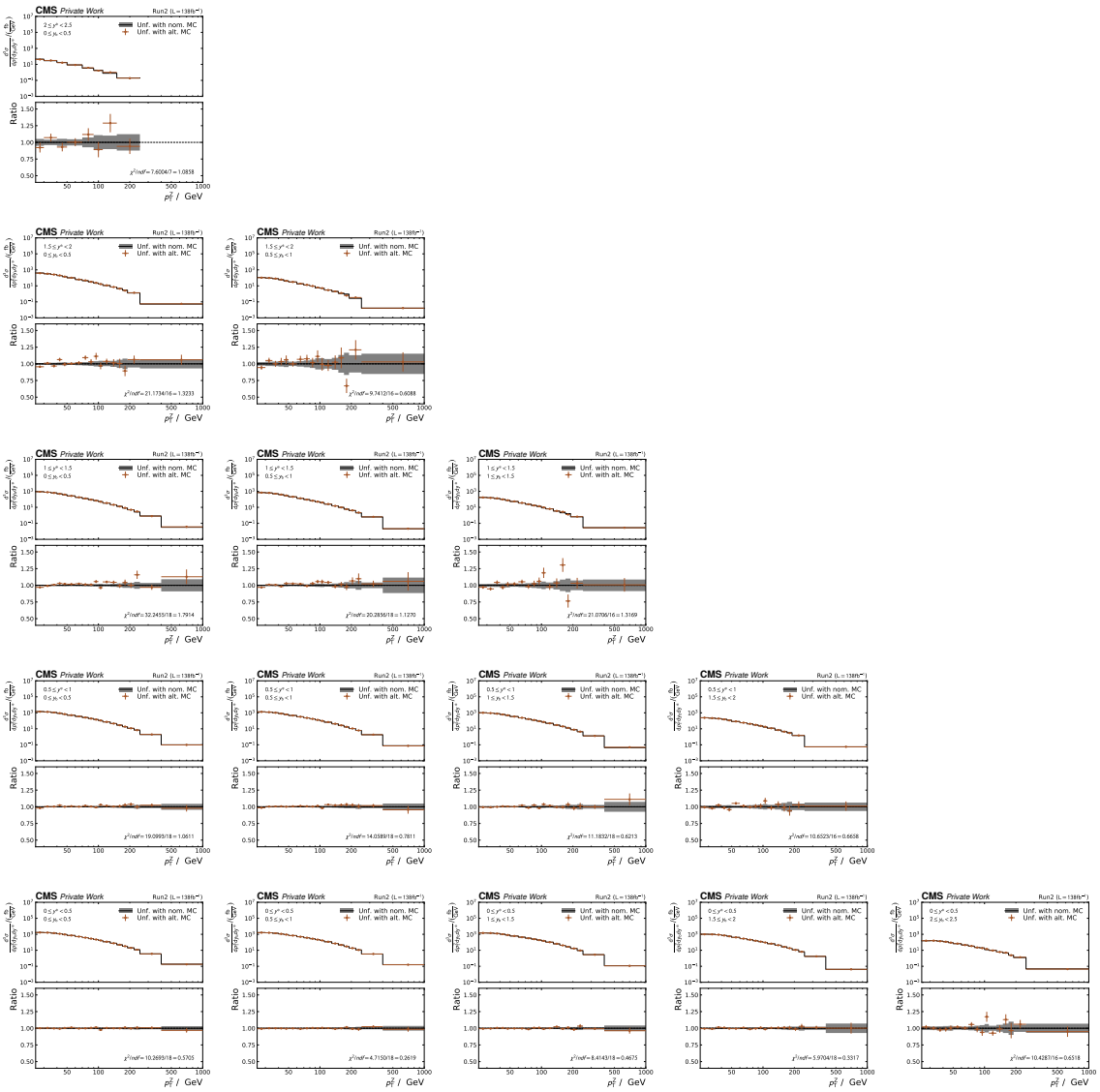


Figure A.22: Check of the systematic bias introduced by choice of a specific simulation in the unfolding procedure used for the combined Run 2 data. Two migration matrices are constructed from the combined set of simulated events from two distinct generators. The two sets of unfolded cross sections obtained from using the two alternative migration matrices are compared. The results obtained with statistical uncertainties from the nominal simulation used in this analysis are shown as a gray band. The results from the alternative are shown as orange points with whiskers showing the corresponding statistical uncertainties. No significant deviations between the two sets are observed apart from statistical fluctuations.

A.4 Uncertainties

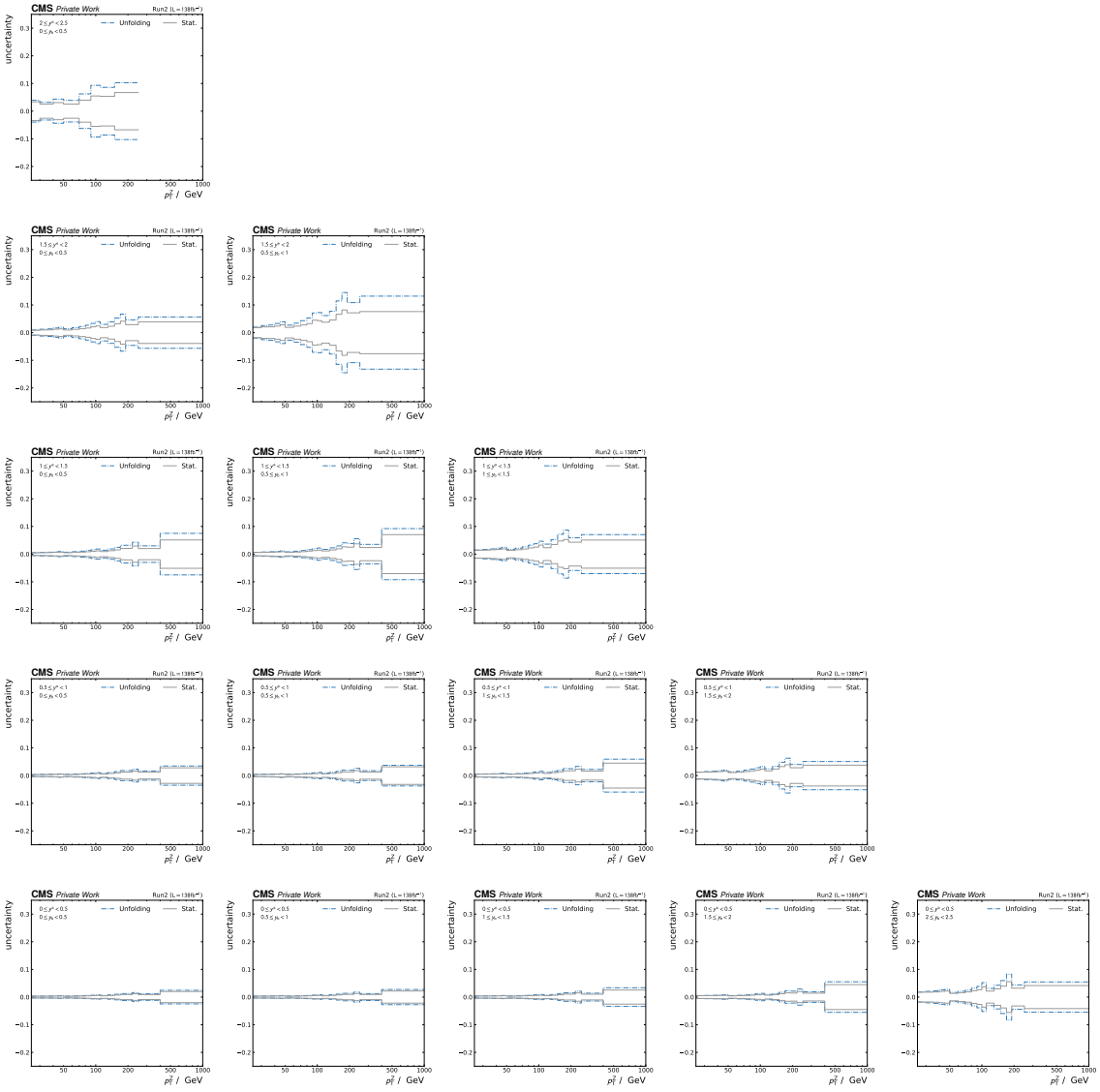


Figure A.23: Uncertainties originating in the limited statistics in data (statistical uncertainty) (gray) and the limited statistics in simulation utilised for the construction of the migration matrix (unfolding) (blue) for the unfolded cross sections obtained for the combined Run 2 data. Both uncertainties are derived by the TUnfold package. Since the number of events are the smallest for high p_T^Z, y^*, and y_b in both, data and simulation, the uncertainties are largest for high p_T^Z and rapidities. They dominate in these regions of the analysed phase space.

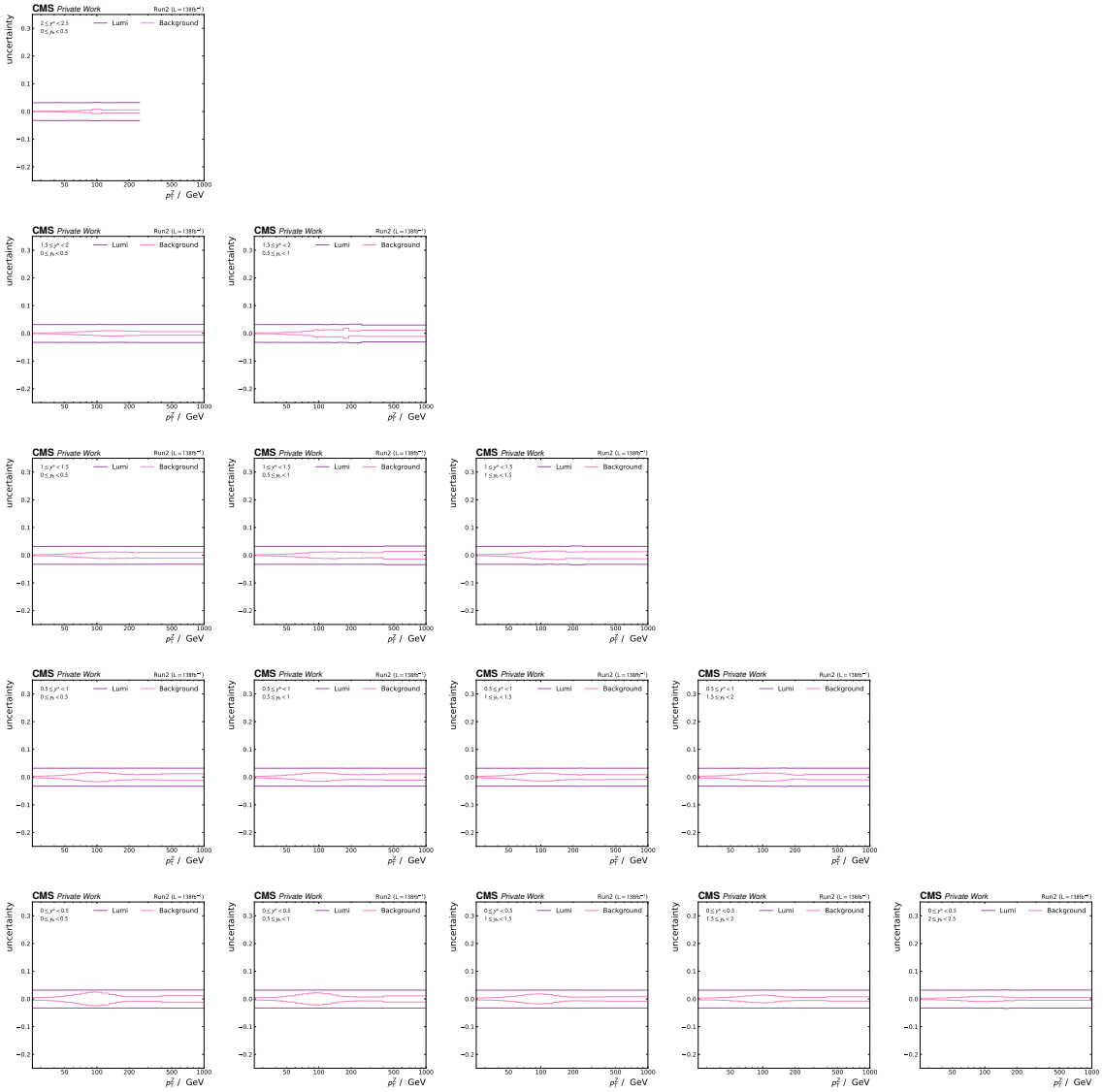


Figure A.24: Background (pink) and luminosity (violet) uncertainties for the unfolded cross sections obtained for the combined Run 2 data. The luminosity uncertainty is derived by varying the nominal unfolded result by the combined luminosity uncertainty of 1.6% taking all correlations into account. It is the same over the whole phase space. The background uncertainty is derived from varying the background contributions subtracted from the measured data yields and propagating each variation through the unfolding. It is largest for p_T^Z close to the mass of the Z boson, where the background contribution is the largest but always smaller than the luminosity uncertainty.

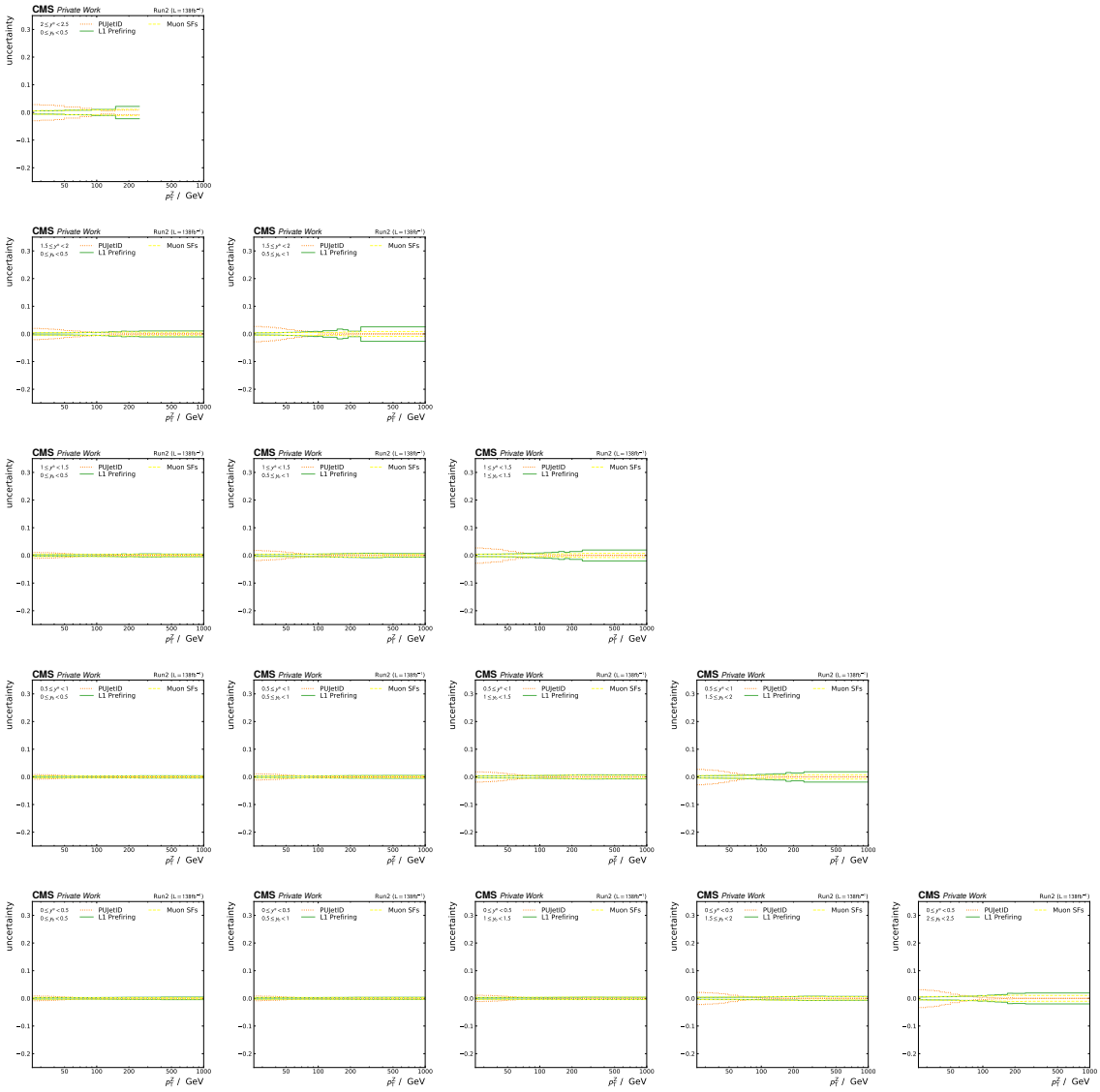


Figure A.25: Muon scale factor (yellow), L1 prefiring (green), and PU jet identification (orange) uncertainties for the unfolded cross sections obtained for the combined Run 2 data. They are derived by constructing alternative migration matrices from the events reconstructed with variations of the muon scale factors, the L1 prefiring correction weights, and the PUJetID efficiency correction weights, respectively within their corresponding uncertainties. For each the unfolding of the measured data yields is repeated and the difference between the nominal and the alternative unfolded cross sections is interpret as the corresponding uncertainty. The PUJetID uncertainty contributes mostly in the low p_T^Z region and decreases towards high p_T^Z. The L1 prefiring uncertainty increases with p_T^Z. The muon scale factor uncertainty has only a slight dependence on p_T^Z. They are significantly smaller than the luminosity uncertainty in all analysed bins except for the PUJetID uncertainty which reaches similar orders of magnitude for the smallest p_T^Z and high rapidities.

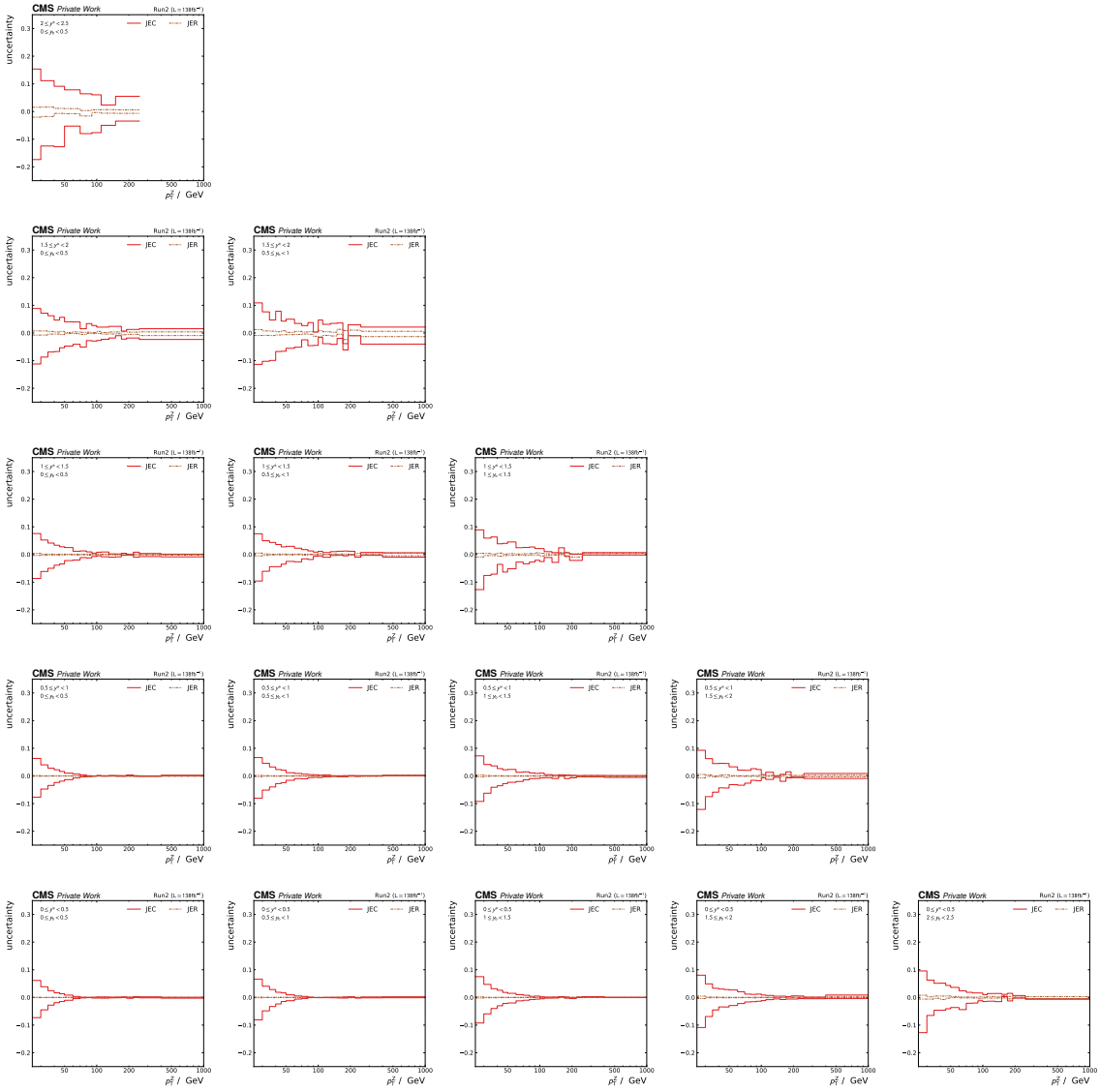


Figure A.26: Jet energy resolution (JER) (brown) and jet energy scale (JEC) (red) uncertainties for the unfolded cross sections obtained for the combined Run 2 data. They are derived by constructing alternative migration matrices from the events with jet energies corrected with scale factors varied respectively within their corresponding uncertainties. Since the JER is assumed to be fully uncorrelated between data-taking periods the variations of each of the four periods are exclusively leading to total four times two variations. For each the unfolding of the measured data yields is repeated and the difference between the nominal and the alternative unfolded cross sections is interpret as the corresponding uncertainty. The total JER uncertainty is constructed as the quadratic sum of the four contributions in each bin. The JER uncertainty contributes the most in the low p_T^Z region and decreases towards high p_T^Z. The JEC uncertainty shows the same behaviour but is an order of magnitude larger. While the JER uncertainty is significantly smaller than the luminosity uncertainty in all analysed bins the JEC uncertainty dominates for small p_T^Z.

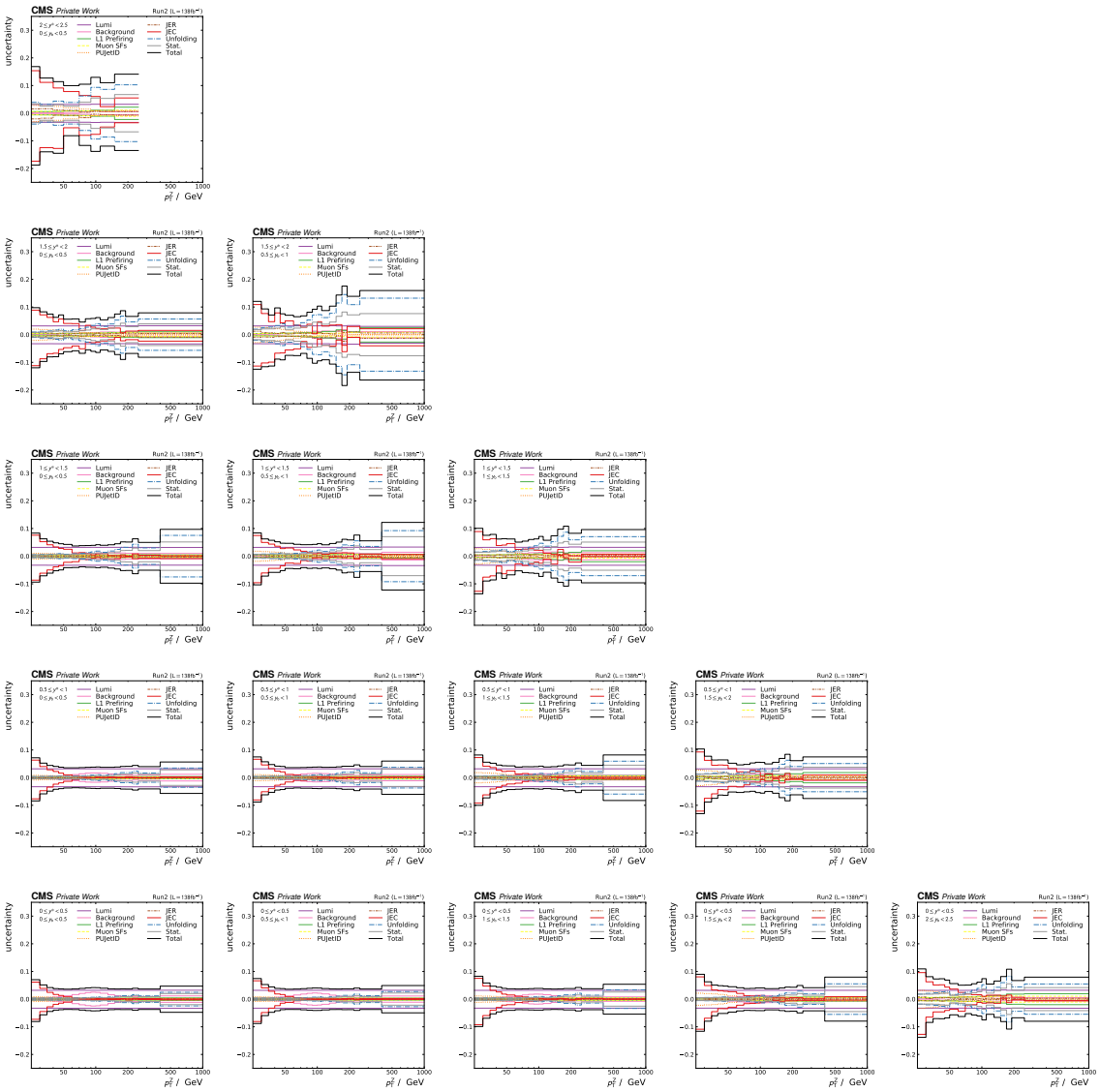


Figure A.27: Overview of all considered uncertainties including uncertainties originating in the limited statistics in data (statistical uncertainty) (gray), originating in the limited statistics in simulation utilised for the construction of the migration matrix (unfolding) (blue), muon scale factor (yellow), L1 prefiring (green), PU jet identification (orange), jet energy resolution (JER) (brown), and jet energy scale (JEC) (red) uncertainties for the unfolded cross sections obtained for the combined Run 2 data. The total uncertainty (black) is defined as the quadratic sum of each individual source's contribution. The low p_T^Z region of phase space is dominated by the JEC uncertainties. In the high p_T^Z region the statistical and unfolding uncertainty dominate.

A.5 Results

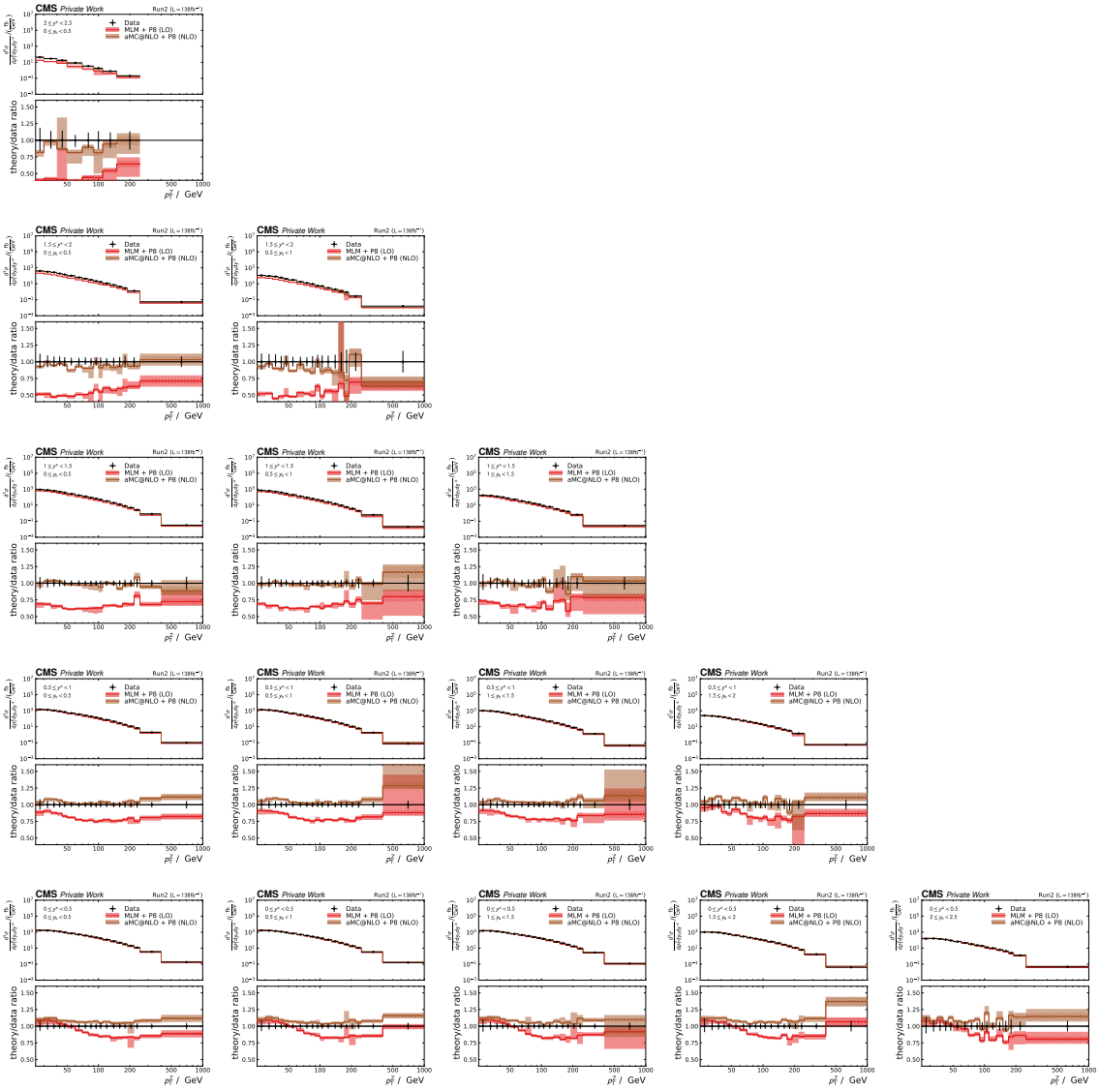


Figure A.28: Measured cross sections corrected for detector effects (black) are compared to theoretical predictions at LO (red) and NLO (brown) accuracy in QCD. The uncertainties on the measured cross sections are the total uncertainties as defined in section 5.6. The uncertainties on the theoretical uncertainties include statistical uncertainties and parton shower uncertainties as defined in section 2.2.5.

APPENDIX B

Computing Simulation Configurations

B.1 Workload Configurations

B.1.1 Scaling Workload

Table B.1: Overview of the workload configuration for the study of the computational complexity of the simulator. The values have been chosen to approximately match the benchmark workload used in [219] with a spread of 10%.

Quantity	Distribution	Mean/Value	Standard Deviation
# Req. CPU cores	–	1	–
FLOP	Gaussian	2164428 M	216442.8 M
Memory	–	2 GB	–
# Input files	–	10	–
Input-file size	Gaussian	3.6 GB	360 MB
# Output files	–	1	–
Output-file size	Gaussian	18 GB	1.8 GB

B.1.2 CMS workloads

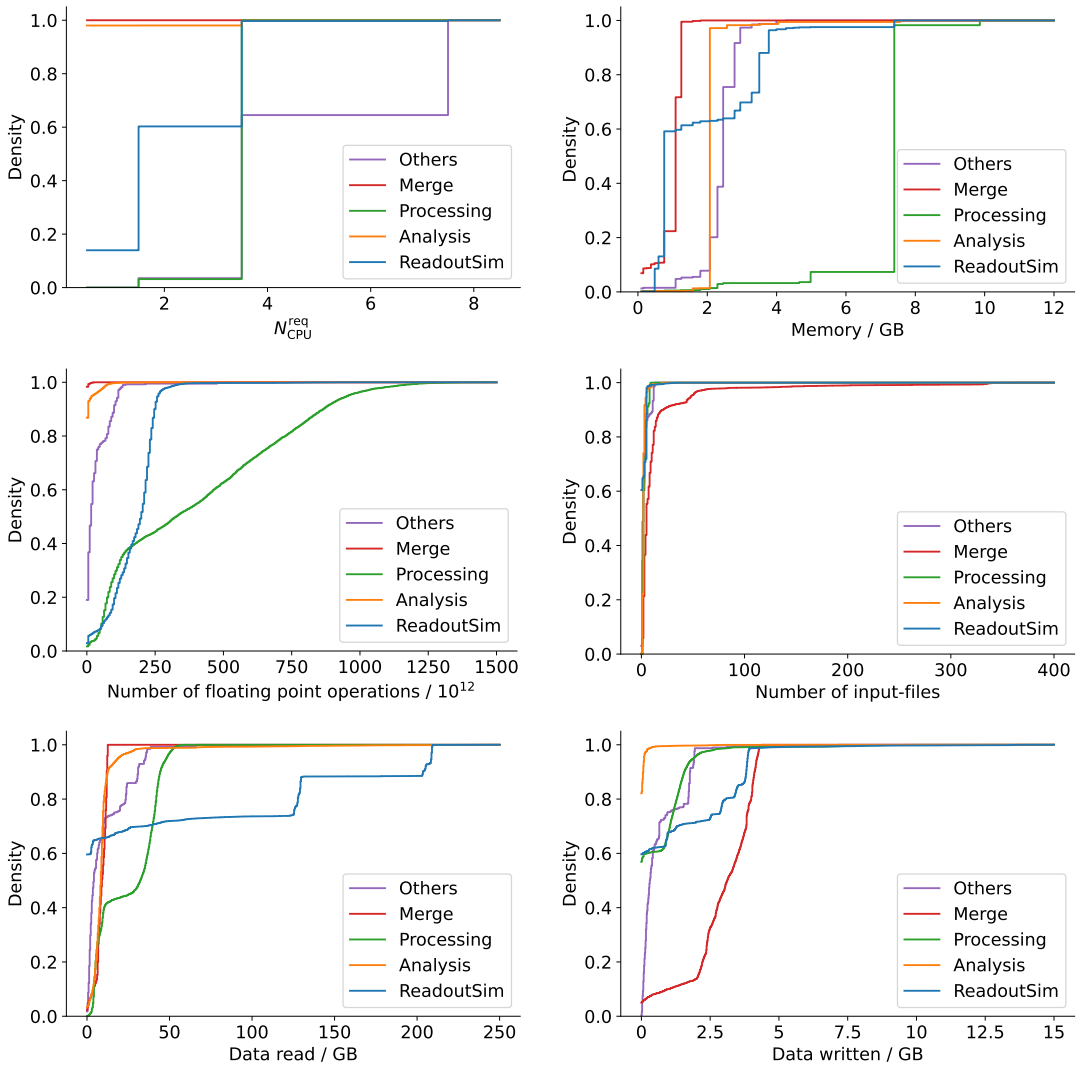


Figure B.1: Cumulative distributions of the job characteristics for each class of jobs executed by the CMS collaboration on the tier 1 centre at KIT from 24th of February to the 7th of March 2023. The distributions for the number of requested CPU cores $N^{\text{req}}_{\text{CPU}}$, the required memory, the reconstructed number of floating point operations, the numbe rof input files and the amount of data read and written by jobs of each of the five classes is shown.

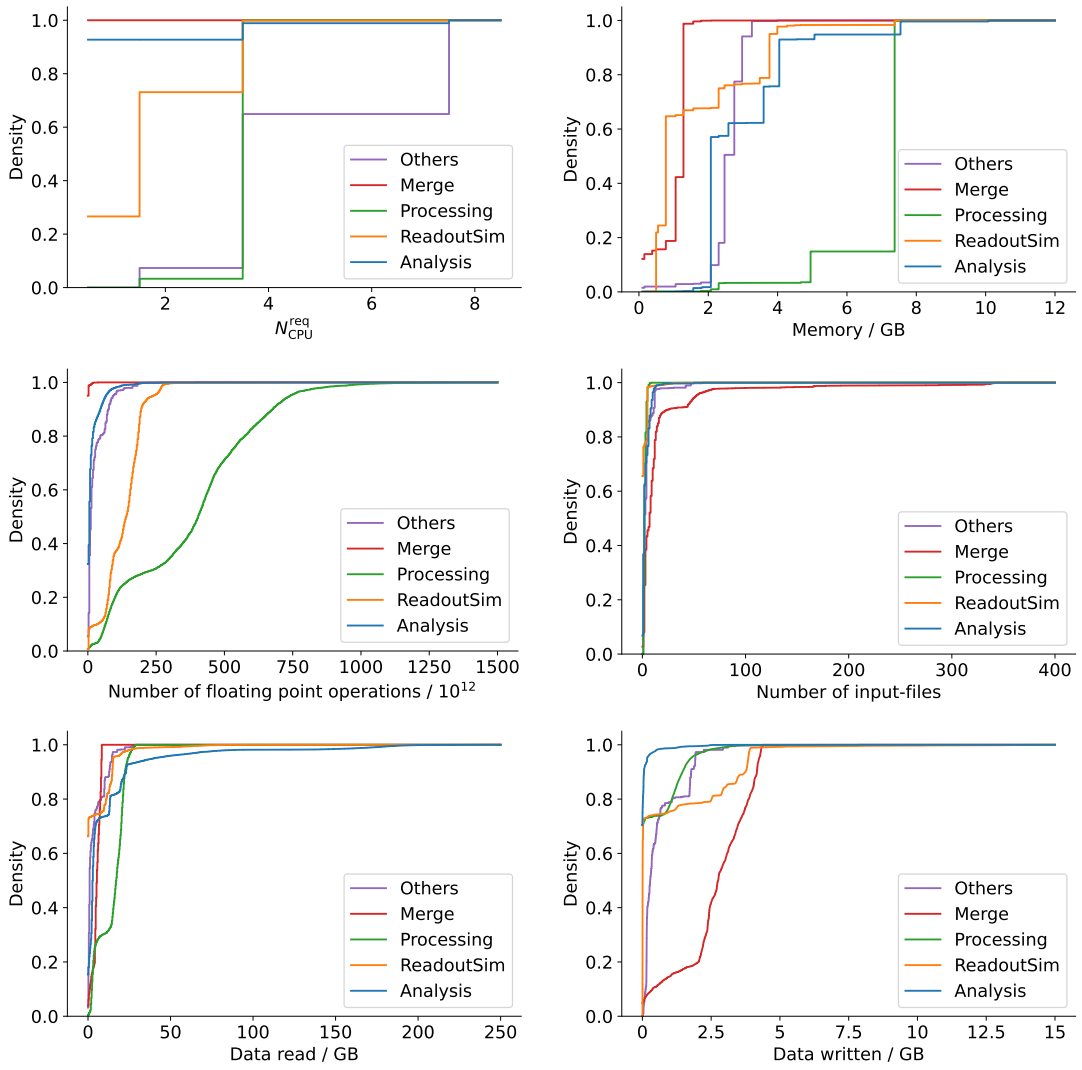


Figure B.2: Cumulative distributions of the job characteristics for each class of jobs executed by the CMS collaboration on the tier 2 centre at DESY from 24th of February to the 7th of March 2023. The distributions for the number of requested CPU cores $N_{\mathrm{CPU}}^{\mathrm{req}}$, the required memory, the reconstructed number of floating point operations, the numbe rof input files and the amount of data read and written by jobs of each of the five classes is shown.

B.2 Platform Configurations

B.2.1 Validation Platform

```
<?xml version="1.0"?>
<platform version="4.1">
<config>
    <prop id="network/loopback-bw" value="1000000000000"/>
</config>

<zone id="global" routing="Full">
    <zone id="ETP" routing="Floyd">
        <host id="sg01.etp.kit.edu" speed="1970Mf" core="24">
            <prop id="type" value="worker,cache"/>
            <prop id="ram" value="64GiB"/>
            <disk id="ssd_cache1" read_bw="17MBps" write_bw="17MBps">
                <prop id="size" value="2TB"/>
                <prop id="mount" value="/"/>
            </disk>
        </host>
        <host id="sg02.etp.kit.edu" speed="1969.583Mf" core="24">
            <prop id="type" value="networkmonitor"/>
            <prop id="ram" value="64GiB"/>
            <disk id="ssd_cache1" read_bw="0.12GBps"
            write_bw="0.12GBps">
                <prop id="size" value="2TB"/>
                <prop id="mount" value="/"/>
            </disk>
        </host>
        <host id="sg03.etp.kit.edu" speed="1990Mf" core="12">
            <prop id="type" value="worker,cache"/>
            <prop id="ram" value="32GiB"/>
            <disk id="ssd_cache1" read_bw="13MBps" write_bw="13MBps">
                <prop id="size" value="2TB"/>
                <prop id="mount" value="/"/>
            </disk>
        </host>
        <host id="sg04.etp.kit.edu" speed="1950Mf" core="12">
            <prop id="type" value="worker,cache"/>
            <prop id="ram" value="32GiB"/>
            <disk id="ssd_cache1" read_bw="9MBps" write_bw="9MBps">
                <prop id="size" value="2TB"/>
                <prop id="mount" value="/"/>
            </disk>
```

```
    </host>
    <host id="WMSHost" speed="10Gf" core="10">
        <prop id="type" value="scheduler,executor"/>
        <prop id="ram" value="16GB"/>
    </host>

    <router id="etpgateway"/>
    <router id="etpswitch"/>

    <link id="loopback" bandwidth="5000GBps" latency="0us"/>
    <link id="etp_link0" bandwidth="10Gbps" latency="0us"/>
    <link id="etp_link1" bandwidth="10Gbps" latency="0us"/>
    <link id="etp_link2up" bandwidth="1.15Gbps" latency="0us"/>
    <link id="etp_link2down" bandwidth="1.15Gbps" latency="0us"/>
    <link id="etp_link0ut" bandwidth="1.6Gbps" latency="0us"/>
    <link id="etp_link3" bandwidth="10Gbps" latency="0us"/>
    <link id="etp_link4" bandwidth="10Gbps" latency="0us"/>

    <route src="etpswitch" dst="WMSHost">
        <link_ctn id="etp_link0"/>
    </route>
    <route src="etpswitch" dst="sg01.etp.kit.edu">
        <link_ctn id="etp_link1"/>
    </route>
    <route src="etpswitch" dst="sg04.etp.kit.edu">
        <link_ctn id="etp_link4"/>
    </route>
    <route src="etpswitch" dst="sg03.etp.kit.edu">
        <link_ctn id="etp_link3"/>
    </route>
    <route src="etpswitch" dst="sg02.etp.kit.edu">
        <link_ctn id="etp_link2up"/>
    </route>
    <route src="etpgateway" dst="sg02.etp.kit.edu">
        <link_ctn id="etp_link2down"/>
    </route>
</zone>

<zone id="Remote" routing="Full">
    <host id="RemoteStorage" speed="1000Gf" core="10">
        <prop id="type" value="storage"/>
        <disk id="hard_drive" read_bw="40GBps" write_bw="40GBps">
            <prop id="size" value="1PB"/>
            <prop id="mount" value="/"/>
```

```
            </disk>
        </host>

        <link id="etp_to_remote" bandwidth="100Gbps" latency="0us"/>
    </zone>

    <zoneRoute src="ETP" dst="Remote" gw_src="etpgateway"
    gw_dst="RemoteStorage">
        <link_ctn id="etp_to_remote"/>
    </zoneRoute>
</zone>
</platform>
```

B.2.2 Scaled Validation Platform

```
<?xml version="1.0"?>
<platform version="4.1">
<config>
    <prop id="network/loopback-bw" value="1000000000000"/>
</config>

<zone id="global" routing="Full">
    <zone id="ETP" routing="Floyd">
        <host id="sg01.etp.kit.edu" speed="1970Mf" core="240">
            <prop id="type" value="worker,cache"/>
            <prop id="ram" value="64GiB"/>
            <disk id="ssd_cache1" read_bw="170MBps"
            write_bw="170MBps">
                <prop id="size" value="2TB"/>
                <prop id="mount" value="/"/>
            </disk>
        </host>
        <host id="sg02.etp.kit.edu" speed="1969.583Mf" core="24">
            <prop id="type" value="networkmonitor"/>
            <prop id="ram" value="64GiB"/>
            <disk id="ssd_cache1" read_bw="1.2GBps"
            write_bw="1.2GBps">
                <prop id="size" value="2TB"/>
                <prop id="mount" value="/"/>
            </disk>
        </host>
        <host id="sg03.etp.kit.edu" speed="1990Mf" core="120">
            <prop id="type" value="worker,cache"/>
```

```
    <prop id="ram" value="32GiB"/>
    <disk id="ssd_cache1" read_bw="130MBps"
    write_bw="130MBps">
        <prop id="size" value="2TB"/>
        <prop id="mount" value="/"/>
    </disk>
</host>
<host id="sg04.etp.kit.edu" speed="1950Mf" core="120">
    <prop id="type" value="worker,cache"/>
    <prop id="ram" value="32GiB"/>
    <disk id="ssd_cache1" read_bw="90MBps" write_bw="90MBps">
        <prop id="size" value="2TB"/>
        <prop id="mount" value="/"/>
    </disk>
</host>
<host id="WMSHost" speed="10Gf" core="100">
    <prop id="type" value="scheduler,executor"/>
    <prop id="ram" value="16GB"/>
</host>

<router id="etpgateway"/>
<router id="etpswitch"/>

<link id="loopback" bandwidth="5000GBps" latency="0us"/>
<link id="etp_link0" bandwidth="100Gbps" latency="0us"/>
<link id="etp_link1" bandwidth="100Gbps" latency="0us"/>
<link id="etp_link2up" bandwidth="11.5Gbps" latency="0us"/>
<link id="etp_link2down" bandwidth="11.5Gbps" latency="0us"/>
<link id="etp_link0ut" bandwidth="16Gbps" latency="0us"/>
<link id="etp_link3" bandwidth="100Gbps" latency="0us"/>
<link id="etp_link4" bandwidth="100Gbps" latency="0us"/>

<route src="etpswitch" dst="WMSHost">
    <link_ctn id="etp_link0"/>
</route>
<route src="etpswitch" dst="sg01.etp.kit.edu">
    <link_ctn id="etp_link1"/>
</route>
<route src="etpswitch" dst="sg04.etp.kit.edu">
    <link_ctn id="etp_link4"/>
</route>
<route src="etpswitch" dst="sg03.etp.kit.edu">
    <link_ctn id="etp_link3"/>
</route>
```

```
        <route src="etpswitch" dst="sg02.etp.kit.edu">
            <link_ctn id="etp_link2up"/>
        </route>
        <route src="etpgateway" dst="sg02.etp.kit.edu">
            <link_ctn id="etp_link2down"/>
        </route>
    </zone>

    <zone id="Remote" routing="Full">
        <host id="RemoteStorage" speed="1000Gf" core="100">
            <prop id="type" value="storage"/>
            <disk id="hard_drive" read_bw="400GBps" write_bw="400GBps">
                <prop id="size" value="1PB"/>
                <prop id="mount" value="/"/>
            </disk>
        </host>

        <link id="etp_to_remote" bandwidth="1000Gbps" latency="0us"/>
    </zone>

    <zoneRoute src="ETP" dst="Remote" gw_src="etpgateway"
    gw_dst="RemoteStorage">
        <link_ctn id="etp_to_remote"/>
    </zoneRoute>
</zone>
</platform>
```

B.2.3 Diskless Tier 2 Platform

```
<?xml version="1.0"?>
<platform version="4.1">
<config>
    <prop id="network/loopback-bw" value="1000000000000"/>
</config>

<zone id="global" routing="Floyd">

    <zone id="KIT" routing="Floyd">

        <zone id="GridKA" routing="Floyd">

            <cluster id="Tier1" prefix="Tier1" radical="0-9" suffix=""
            speed="2555Mf" core="42" bw="1150Mbps" lat="0us">
```

```
    <prop id="type" value="worker"/>
    <prop id="ram" value="1187.20GiB"/>
</cluster>

<zone id="GridKA-service" routing="Floyd">

    <host id="GridKA_dCache" speed="1000Gf" core="10">
        <prop id="type" value="storage"/>
        <disk id="hard_drive" read_bw="920Mbps"
        write_bw="920Mbps">
            <prop id="size" value="7PB"/>
            <prop id="mount" value="/"/>
        </disk>
    </host>

    <host id="WMSHost" speed="10Gf" core="10">
        <prop id="type" value="scheduler,executor"/>
        <prop id="ram" value="16GB"/>
    </host>

    <router id="GridKAgateway"/>

    <link id="GridKA_sched"
    bandwidth="115Mbps" latency="0us"/>

    <link id="GridKA_Tier1_FATPIPE" bandwidth="1150Mbps"
    latency="0us" sharing_policy="FATPIPE"/>
    <link id="GridKA_Tier1"
    bandwidth="2300Mbps" latency="0us"/>

    <link id="GridKA_dcachepool_FATPIPE" bandwidth="460Mbps"
    latency="0us" sharing_policy="FATPIPE"/>
    <link id="GridKA_dcachepool"
    bandwidth="920Mbps" latency="0us"/>

    <route src="GridKAgateway" dst="WMSHost">
        <link_ctn id="GridKA_sched"/>
    </route>

    <route src="GridKAgateway" dst="GridKA_dCache">
        <link_ctn id="GridKA_dcachepool_FATPIPE"/>
        <link_ctn id="GridKA_dcachepool"/>
    </route>
</zone>
```

```
            <zoneRoute src="GridKA-service" dst="Tier1"
            gw_src="GridKAgateway" gw_dst="Tier1Tier1_router">
                <link_ctn id="GridKA_Tier1_FATPIPE"/>
                <link_ctn id="GridKA_Tier1"/>
            </zoneRoute>

        </zone>

        <zone id="KITcentral" routing="Floyd">

            <router id="KITgateway"/>

            <link id="GridKA_to_KIT" bandwidth="1150Mbps"
            latency="0us"/>
            <link id="KIT_to_DESY" bandwidth="1150Mbps" latency="0us">
                <prop id="variation" value="100000000,50000000"/>
            </link>

        </zone>

        <zoneRoute src="GridKA" dst="KITcentral"
        gw_src="GridKAgateway" gw_dst="KITgateway">
            <link_ctn id="GridKA_to_KIT"/>
        </zoneRoute>

    </zone>

    <zone id="DESY" routing="Floyd">

        <zone id="DESYGrid" routing="Floyd">

            <!-- <host id="Tier2" speed="2761Mf" core="200"> -->
            <host id="Tier2" speed="2209Mf" core="200">
                <prop id="type" value="worker"/>
                <prop id="ram" value="500GiB"/>
            </host>

            <host id="DESY_dCache" speed="1000Gf" core="10">
                <prop id="type" value="cache"/>
                <disk id="hard_drive"
                read_bw="920Mbps" write_bw="920Mbps">
                    <prop id="size" value="7PB"/>
                    <prop id="mount" value="/"/>
```

```
            </disk>
        </host>

        <router id="DESYGridgateway"/>

        <link id="DESY_Tier2" bandwidth="460Mbps" latency="0us"/>

        <link id="DESY_dCachepool" bandwidth="460Mbps"
        latency="0us"/>

        <route src="DESYGridgateway" dst="Tier2">
            <link_ctn id="DESY_Tier2"/>
        </route>

        <route src="DESYGridgateway" dst="DESY_dCache">
            <link_ctn id="DESY_dCachepool"/>
        </route>

    </zone>

    <zone id="DESYcentral" routing="Floyd">

        <router id="DESYgateway"/>

        <link id="DESYGrid_to_DESY"
        bandwidth="1150Mbps" latency="0us"/>

    </zone>

    <zoneRoute src="DESYGrid" dst="DESYcentral"
    gw_src="DESYGridgateway" gw_dst="DESYgateway">
        <link_ctn id="DESYGrid_to_DESY"/>
    </zoneRoute>

</zone>

<zoneRoute src="KIT" dst="DESY"
gw_src="KITgateway" gw_dst="DESYgateway">
    <link_ctn id="KIT_to_DESY"/>
</zoneRoute>

</zone>
</platform>
```

Acronyms

ALICE A Large Ion Collider Experiment.

AOD Analysis Object Data.

APV25 analog pipeline voltage.

ATLAS A Toroidal LHC ApparatuS.

BOOSTER Proton Synchrotron Booster.

BSM Beyond Standard Model.

CA certification authority.

CE compute element.

CERN Organisation européenne pour la recherche nucléaire.

CHS charged hadron subtraction.

CMS Compact Muon Solenoid.

CPU central processing unit.

CSC cathode strip chamber.

CVMFS CERN virtual machine file system.

DAQ data acquisition.

DB direct balance.

DESY Deutsches Elektronen-Synchrotron.

DGLAP Dokshitzer-Gribov-Lipatov-Altarelli-Parisi.

DIGI digitization.

DT drift tube.

EB ECAL barrel.

ECAL electromagnetic calorimeter.

EDM Event Data Model.

EE ECAL endcap.

ES ECAL preshower.

ETP Institute of Experimental Particle Physics.

EW electroweak theory.

FIFO First-In-First-Out.

FLOP floating point operation.

FPGA field-programmable gate array.

FSR final-state radiation.

GCT global calorimeter trigger.

GEN generator.

GMT global muon trigger.

GPU graphics processing unit.

GridKa Grid Computing Centre Karlsruhe.

GT L1 global trigger system.

HB HCAL barrel.

HCAL hadronic calorimeter.

HDD hard drive disk.

HE HCAL endcap.

HEP high energy physics.

HEP-SPEC06 HEP Standard Performance Evaluation Corporation 06.

HF HCAL forward.

HL-LHC High Luminosity Large Hadron Collider.

HLT high level trigger.

HO HCAL outer.

HPC high-performance computing.

HTTP hypertext transfer protocol.

IP interaction point.

IRC infrared and collinear.

ISR initial-state radiation.

JDL job description language.

JEC jet energy calibration.

JERC *jet energy resolution and correction*.

JSON JavaScript Object Notation.

JWT JSON web token.

KIT Karlsruhe Institute of Technology.

L1 level 1 trigger.

L1A L1 accept.

LAN local area network.

LEP Large Electron-Positron Collider.

LHC Large Hadron Collider.

LHCb Large Hadron Collider beauty.

LHCf Large Hadron Collider forward.

LHCONE LHC Open Network Environment.

LHCOPN LHC Optical Private Network.

LL leading-logarithmic.

LO leading order.

LRMS local resource management system.

LRU Least-Recently-Used.

LSDCS large scale distributed computing systems.

LV leading vertex.

MC Monte Carlo.

ME matrix element.

MET missing transverse energy.

MFA multi-factor authentication.

MINIAOD Mini AOD.

MIP minimum ionizing particle.

MONARC Models of Networked Analysis at Regional Centres.

MPF missing transverse energy projection fraction.

MPI multiple-parton interaction.

N^3LO next-to-next-to-leading order.

NAF National Analysis Facility.

NANOAOD Nano AOD.

NLL next-to-leading-logarithmic.

NLO next-to-leading order.

NNLL next-to-next-to-leading-logarithmic.

NNLO next-to-next-to-leading order.

NP non-perturbative.

OBS overlay batch system.

OS operating system.

OTP one-time password.

PDF parton distribution function.

PF particle flow.

PS parton shower.

PU pileup.

PUJetID PU jet identification.

PUPPI pileup per particle identification.

PV primary vertex.

QCD quantum chromodynamics.

QED quantum electrodynamics.

QFT quantum field theory.

RAM random-access memory.

RCT regional calorimeter trigger.

RECO reconstruction.

RHEL Red Hat Enterprise Linux.

RNG random number generator.

RPC resistive plate chamber.

RTT round trip time.

Rucio Rucio.

RWTH RWTH Aachen University.

SIM simulation.

SM Standard Model.

SPS Super Proton Synchrotron.

TCS trigger control system.

TEC tracker end caps.

TIB tracker inner barrel.

TID tracker inner disk.

TOB tracker outer barrel.

TOpAS Throughput Optimized Analysis System.

TOTEM Total Elastic and Diffractive Cross Section Measurement.

TP trigger primitive.

UE underlying event.

UHH Universität Hamburg.

VFP preamplifier feedback voltage bias.

VM virtual machine.

VO virtual organization.

VOMS virtual organization membership service.

WAN wide area network.

WLCG Worldwide LHC Computing Grid.

WMS Workload Management System.

List of Figures

List of Tables

References

[1] R. L. Workman et al. „Review of Particle Physics“. *Progress of Theoretical and Experimental Physics* 2022.8 (Aug. 2022).
DOI: 10.1093/ptep/ptac097.

[2] M. E. Peskin. „An introduction to quantum field theory“. Addison-Wesley Pub. Co., 1995, p. 842. ISBN: 0201503972.

[3] D. J. Griffiths. „Introduction to elementary particles“. Wiley, 1987, p. 392. ISBN: 0471603864.

[4] S. Weinberg. „The Quantum Theory of Fields“. Vol. 1. Cambridge University Press, June 1995.
DOI: 10.1017/cbo9781139644167.

[5] S. Weinberg. „The Quantum Theory of Fields“. Vol. 2. Cambridge University Press, Aug. 1996.
DOI: 10.1017/cbo9781139644174.

[6] V. Bargmann and E. P. Wigner. „Group Theoretical Discussion of Relativistic Wave Equations“. *Proceedings of the National Academy of Sciences* 34.5 (May 1948), pp. 211–223.
DOI: 10.1073/pnas.34.5.211.

[7] W. Pauli. „The Connection Between Spin and Statistics“. *Phys. Rev.* 58 (8 Oct. 1940), pp. 716–722.
DOI: 10.1103/PhysRev.58.716.

[8] M. Gell-Mann. „Symmetries of Baryons and Mesons“. *Phys. Rev.* 125 (3 Feb. 1962), pp. 1067–1084.
DOI: 10.1103/PhysRev.125.1067.

[9] C. S. Wu et al. „Experimental Test of Parity Conservation in Beta Decay“. *Phys. Rev.* 105 (4 Feb. 1957), pp. 1413–1415.
DOI: 10.1103/PhysRev.105.1413.

[10] F. Hasert et al. „Observation of neutrino-like interactions without muon or electron in the gargamelle neutrino experiment“. *Physics Letters B* 46.1 (1973), pp. 138–140. ISSN: 0370-2693.
DOI: https://doi.org/10.1016/0370-2693(73)90499-1.

[11] S. L. Glashow. „Partial-symmetries of weak interactions“. *Nuclear Physics* 22.4 (1961), pp. 579–588. ISSN: 0029-5582.
DOI: https://doi.org/10.1016/0029-5582(61)90469-2.

[12] S. Weinberg. „A Model of Leptons“. *Phys. Rev. Lett.* 19 (21 Nov. 1967), pp. 1264–1266.
DOI: 10.1103/PhysRevLett.19.1264.

[13] A. Salam. „Weak and Electromagnetic Interactions“. *Conf. Proc. C* 680519 (1968), pp. 367–377.
DOI: 10.1142/9789812795915_0034.

[14] F. Englert and R. Brout. „Broken Symmetry and the Mass of Gauge Vector Mesons“. *Phys. Rev. Lett.* 13 (9 Aug. 1964), pp. 321–323.
DOI: 10.1103/PhysRevLett.13.321.

[15] P. Higgs. „Broken symmetries, massless particles and gauge fields“. *Physics Letters* 12.2 (1964), pp. 132–133. ISSN: 0031-9163.
DOI: https://doi.org/10.1016/0031-9163(64)91136-9.

[16] P. W. Higgs. „Broken Symmetries and the Masses of Gauge Bosons“. *Phys. Rev. Lett.* 13 (16 Oct. 1964), pp. 508–509.
DOI: 10.1103/PhysRevLett.13.508.

[17] P. W. Higgs. „Spontaneous Symmetry Breakdown without Massless Bosons“. *Phys. Rev.* 145 (4 May 1966), pp. 1156–1163.
DOI: 10.1103/PhysRev.145.1156.

[18] R. P. Feynman. „Space-Time Approach to Quantum Electrodynamics“. *Phys. Rev.* 76 (6 Sept. 1949), pp. 769–789.
DOI: 10.1103/PhysRev.76.769.

[19] F. James. „A review of pseudorandom number generators“. *Computer Physics Communications* 60.3 (1990), pp. 329–344. ISSN: 0010-4655.
DOI: https://doi.org/10.1016/0010-4655(90)90032-V.

[20] G. Marsaglia, B. Narasimhan, and A. Zaman. „A random number generator for PC's“. *Computer Physics Communications* 60.3 (1990), pp. 345–349. ISSN: 0010-4655.
DOI: https://doi.org/10.1016/0010-4655(90)90033-W.

[21] J. C. Collins, D. E. Soper, and G. Sterman. „Factorization of Hard Processes in QCD“. *Perturbative QCD*. World Scientific, July 1989, pp. 1–91.
DOI: 10.1142/9789814503266_0001.

[22] A. Buckley et al. „General-purpose event generators for LHC physics“. *Physics Reports* 504.5 (2011), pp. 145–233. ISSN: 0370-1573.
DOI: https://doi.org/10.1016/j.physrep.2011.03.005.

[23] V. V. Sudakov. „Vertex parts at very high-energies in quantum electrodynamics“. *Sov. Phys. JETP* 3 (1956), pp. 65–71.

[24] G. Altarelli and G. Parisi. „Asymptotic freedom in parton language“. *Nuclear Physics B* 126.2 (1977), pp. 298–318. ISSN: 0550-3213.
DOI: https://doi.org/10.1016/0550-3213(77)90384-4.

[25] T. Sjöstrand. „A model for initial state parton showers“. *Physics Letters B* 157.4 (1985), pp. 321–325. ISSN: 0370-2693.
DOI: https://doi.org/10.1016/0370-2693(85)90674-4.

[26] G. Marchesini and B. Webber. „Monte Carlo simulation of general hard processes with coherent QCD radiation“. *Nuclear Physics B* 310.3 (1988), pp. 461–526. ISSN: 0550-3213.
DOI: https://doi.org/10.1016/0550-3213(88)90089-2.

[27] J. Bellm et al. „Herwig 7.0/Herwig++ 3.0 release note“. *The European Physical Journal C* 76.4 (Apr. 2016).
DOI: 10.1140/epjc/s10052-016-4018-8.

[28] M. Bähr et al. „Herwig++ physics and manual“. *The European Physical Journal C* 58.4 (Nov. 2008), pp. 639–707.
DOI: 10.1140/epjc/s10052-008-0798-9.

[29] T. Sjöstrand et al. „An introduction to PYTHIA 8.2“. *Computer Physics Communications* 191 (2015), pp. 159–177. ISSN: 0010-4655.
DOI: https://doi.org/10.1016/j.cpc.2015.01.024.

[30] S. Frixione and B. R. Webber. „Matching NLO QCD computations and parton shower simulations“. *Journal of High Energy Physics* 2002.06 (June 2002), pp. 029–029.
DOI: 10.1088/1126-6708/2002/06/029.

[31] P. Nason. „A New Method for Combining NLO QCD with Shower Monte Carlo Algorithms“. *Journal of High Energy Physics* 2004.11 (Nov. 2004), pp. 040–040.
DOI: 10.1088/1126-6708/2004/11/040.

[32] B. Andersson et al. „Parton fragmentation and string dynamics“. *Physics Reports* 97.2-3 (July 1983), pp. 31–145.
DOI: 10.1016/0370-1573(83)90080-7.

[33] B. Webber. „A QCD model for jet fragmentation including soft gluon interference“. *Nuclear Physics B* 238.3 (June 1984), pp. 492–528.
DOI: 10.1016/0550-3213(84)90333-x.

[34] B. Andersson, G. Gustafson, and B. Soderberg. „A General Model for Jet Fragmentation“. *Z. Phys. C* 20 (1983), p. 317.
DOI: 10.1007/BF01407824.

[35] D. Amati and G. Veneziano. „Preconfinement as a Property of Perturbative QCD“. *Phys. Lett. B* 83 (1979), pp. 87–92. ISSN: 0370-2693.
DOI: 10.1016/0370-2693(79)90896-7.

[36] CMS Collaboration. „Development and validation of HERWIG 7 tunes from CMS underlying-event measurements“. *Eur. Phys. J. C* 81.4 (2021), p. 312. DOI: 10.1140/epjc/s10052-021-08949-5. arXiv: 2011.03422.

[37] CMS Collaboration. „Extraction and validation of a new set of CMS PYTHIA8 tunes from underlying-event measurements“. *Eur. Phys. J. C* 80.1 (2020), p. 4. DOI: 10.1140/epjc/s10052-019-7499-4. arXiv: 1903.12179 [hep-ex].

[38] DELPHI Collaboration. „Tuning and test of fragmentation models based on identified particles and precision event shape data“. *Z. Phys. C* 73 (1996), pp. 11–60. DOI: 10.1007/s002880050295.

[39] A. Buckley et al. „Systematic event generator tuning for the LHC“. *The European Physical Journal C* 65.1-2 (Nov. 2009), pp. 331–357. DOI: 10.1140/epjc/s10052-009-1196-7.

[40] D. J. Lange et al. „Upgrades for the CMS simulation“. *Journal of Physics: Conference Series* 608 (May 2015), p. 012056. DOI: 10.1088/1742-6596/608/1/012056.

[41] S. Agostinelli et al. „Geant4a simulation toolkit“. *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.

[42] J. Gao et al. „CT10 next-to-next-to-leading order global analysis of QCD“. *Physical Review D* 89.3 (Feb. 2014), p. 033009. DOI: 10.1103/physrevd.89.033009.

[43] R. D. Ball et al. „Parton distributions from high-precision collider data“. *The European Physical Journal C* 77.10 (Oct. 2017). DOI: 10.1140/epjc/s10052-017-5199-5.

[44] V. N. Gribov and L. N. Lipatov. „Deep inelastic e p scattering in perturbation theory“. *Sov. J. Nucl. Phys.* 15 (1972), pp. 438–450.

[45] H. David Politzer. „Asymptotic freedom: An approach to strong interactions“. *Physics Reports* 14.4 (1974), pp. 129–180. ISSN: 0370-1573. DOI: https://doi.org/10.1016/0370-1573(74)90014-3.

[46] O. S. Brüning et al. „LHC Design Report“. CERN Yellow Reports: Monographs. Geneva: CERN, 2004. DOI: 10.5170/CERN-2004-003-V-1.

[47] LEP Collaboration. „LEP design report“. Report. Copies shelved as reports in LEP, PS and SPS libraries. Geneva: CERN, 1984.

[48] CMS Collaboration. „The CMS experiment at the CERN LHC“. *JINST* 3 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[49] ATLAS Collaboration. „ATLAS detector and physics performance: Technical Design Report, 1“. Technical design report. ATLAS. Geneva: CERN, 1999.

[50] LHCb Collaboration. „LHCb reoptimized detector design and performance: Technical Design Report“. Technical design report. LHCb. Geneva: CERN, 2003.

[51] ALICE Collaboration. „ALICE: Technical proposal for a Large Ion collider Experiment at the CERN LHC“. LHC technical proposal. Geneva: CERN, 1995.

[52] S. van der Meer. „Calibration of the effective beam height in the ISR“. Tech. rep. Geneva: CERN, 1968.

[53] CMS Collaboration. „Pileup mitigation at CMS in 13 TeV data“. *JINST* 15 (2020), P09018.
DOI: 10.1088/1748-0221/15/09/P09018. arXiv: 2003.00503 [hep-ex].

[54] D. Fournier and T. Virdee. „The ATLAS and CMS detectors at the LHC“. *Comptes Rendus Physique* 16.4 (2015). Highlights of the LHC run 1 / Résultats marquants de la première période d'exploitation du GCH, pp. 356–367. ISSN: 1631-0705.
DOI: https://doi.org/10.1016/j.crhy.2015.03.018.

[55] M. Ressegotti and O. behalf of the CMS Collaboration. „Overview of the CMS Detector Performance at LHC Run 2“. *Universe* 5.1 (2019). ISSN: 2218-1997.
DOI: 10.3390/universe5010018.

[56] CMS Collaboration. „The CMS tracker system project: Technical Design Report“. Technical design report. CMS. Geneva: CERN, 1997.

[57] CMS Collaboration. „The CMS Phase-1 Pixel Detector Upgrade“. *JINST* 16.02 (2021), P02027.
DOI: 10.1088/1748-0221/16/02/P02027. arXiv: 2012.14304 [physics.ins-det].

[58] „The CMS electromagnetic calorimeter project : Technical Design Report“. Technical design report. CMS. Geneva: CERN, 1997.

[59] „The CMS hadron calorimeter project : Technical Design Report“. Technical design report. CMS. Geneva: CERN, 1997.

[60] CMS Collaboration. „The CMS muon project: Technical Design Report“. Technical design report. CMS. Geneva: CERN, 1997.

[61] CMS Collaboration. „CMS muon system towards LHC Run 2 and beyond“. Tech. rep. Geneva: CERN, 2016.
DOI: 10.1016/j.nuclphysbps.2015.09.159.

[62] CMS Collaboration. „The CMS trigger system“. *JINST* 12.01 (2017), P01020.
DOI: 10.1088/1748-0221/12/01/P01020. arXiv: 1609.02366 [physics.ins-det].

[63] CMS Collaboration. „CMS Technical Design Report for the Level-1 Trigger Upgrade“. Tech. rep. Geneva, 2013.

[64] CMS Collaboration. „Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ TeV“. *JINST* 15.10 (2020), P10017.
DOI: 10.1088/1748-0221/15/10/P10017. arXiv: 2006.10165 [hep-ex].

[65] CMS Collaboration Collaboration. „CMS The TriDAS Project: Technical Design Report, Volume 2: Data Acquisition and High-Level Trigger. CMS trigger and data-acquisition project“. Tech. rep. Geneva, 2002.

[66] CMS Collaboration. „Performance of the CMS muon trigger system in proton-proton collisions at $\sqrt{s} = 13$ TeV“. *JINST* 16 (2021), P07001. DOI: 10.1088/1748-0221/16/07/P07001. arXiv: 2102.04790 [hep-ex].

[67] CMS Collaboration. „Description and performance of track and primary-vertex reconstruction with the CMS tracker“. *JINST* 9.10 (2014), P10009. DOI: 10.1088/1748-0221/9/10/P10009. arXiv: 1405.6569 [physics.ins-det].

[68] CMS Collaboration. „CMS track reconstruction performance during Run 2 and developments for Run 3“. *PoS* ICHEP2020 (2021), p. 733. DOI: 10.22323/1.390.0733. arXiv: 2012.07035 [physics.ins-det].

[69] R. Frühwirth. „Application of Kalman filtering to track and vertex fitting“. *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.

[70] CMS Collaboration. „Technical proposal for the Phase-II upgrade of the Compact Muon Solenoid“. CMS Technical Proposal CERN-LHCC-2015-010, CMS-TDR-15-02. 2015.

[71] CMS Collaboration. „Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC“. *JINST* 16 (2021), P05014. DOI: 10.1088/1748-0221/16/05/P05014. arXiv: 2012.06888 [hep-ex].

[72] CMS Collaboration. „Particle-flow reconstruction and global event description with the CMS detector“. *JINST* 12.10 (2017), P10003. DOI: 10.1088/1748-0221/12/10/P10003. arXiv: 1706.04965 [physics.ins-det].

[73] CMS Collaboration. „Performance of CMS Muon Reconstruction in pp Collision Events at $\sqrt{s} = 7$ TeV“. *JINST* 7 (2012), P10002. DOI: 10.1088/1748-0221/7/10/P10002. arXiv: 1206.4071 [physics.ins-det].

[74] CMS Collaboration. „Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV“. *JINST* 13.06 (2018), P06015. DOI: 10.1088/1748-0221/13/06/P06015. arXiv: 1804.04528 [physics.ins-det].

[75] A. Bodek et al. „Extracting Muon Momentum Scale Corrections for Hadron Collider Experiments“. *Eur. Phys. J. C* 72 (2012), p. 2194. DOI: 10.1140/epjc/s10052-012-2194-8. arXiv: 1208.3710 [hep-ex].

[76] G. P. Salam. „Towards Jetography“. *Eur. Phys. J. C* 67 (2010), pp. 637–686. DOI: 10.1140/epjc/s10052-010-1314-6. arXiv: 0906.1833 [hep-ph].

[77] M. Cacciari, G. P. Salam, and G. Soyez. „The anti-k_t jet clustering algorithm“. *JHEP* 04 (2008), p. 063.
DOI: 10.1088/1126-6708/2008/04/063. arXiv: 0802.1189 [hep-ph].

[78] CMS Collaboration. „Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV“. *JINST* 12.02 (2017), P02014.
DOI: 10.1088/1748-0221/12/02/P02014. arXiv: 1607.03663 [hep-ex].

[79] CMS Collaboration. „Jet energy scale and resolution measurement with Run 2 Legacy Data Collected by CMS at 13 TeV“. Tech. rep. 2021.

[80] CMS Collaboration. „Determination of jet energy calibration and transverse momentum resolution in CMS“. *Journal of Instrumentation* 6.11 (Nov. 2011), P11002–P11002.
DOI: 10.1088/1748-0221/6/11/p11002.

[81] CMS Collaboration. „Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC“. *Phys. Lett. B* 716 (2012), pp. 30–61.
DOI: 10.1016/j.physletb.2012.08.021. arXiv: 1207.7235 [hep-ex].

[82] ATLAS Collaboration. „Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC“. *Phys. Lett. B* 716 (2012), pp. 1–29.
DOI: 10.1016/j.physletb.2012.08.020. arXiv: 1207.7214 [hep-ex].

[83] CMS Collaboration. „CMS Collaboration Publishes its 1000th Paper“.
URL: https://cms.cern/news/cms-collaboration-publishes-its-1000th-paper (visited on 09/20/2023).

[84] CMS Collaboration. „The CMS Collaboration“.
URL: https://cms.cern/collaboration (visited on 09/20/2023).

[85] WLCG. „Computing Reasource Information Catalogue: VO requirement list“.
URL: https://wlcg-cric.cern.ch/core/vopledgereq/listcomp/ (visited on 07/03/2023).

[86] European Grid Infrastructure. „EGI High Energy Physics Compute Accounting“.
URL: https://accounting.egi.eu/discipline/High%20energy%20physics/elap_processors/VO/DATE/2022/1/2022/12/ (visited on 07/03/2023).

[87] T. Berger. „Jet energy calibration and triple differential inclusive cross section measurements with Z (→μμ) + jet events at 13 TeV recorded by the CMS detector“. PhD thesis. Karlsruher Institut für Technologie (KIT), 2019. 139 pp.
DOI: 10.5445/IR/1000104286.

[88] M. Schnepf. „Dynamic Provision of Heterogeneous Computing Resources for Computation- and Data-intensive Particle Physics Analyses“. PhD thesis. Karlsruher Institut für Technologie (KIT), 2022. 129 pp.
DOI: 10.5445/IR/1000143165.

[89] C. Verstege. „Measurement of the Triple-Differential Cross- Section of Z+Jet Production with the CMS Detector at 13 TeV“. MA thesis. Karlsruhe Institute of Technology (KIT), 2022.

[90] M. J. Schnepf. „Dynamic Provision of Heterogeneous Computing Resources for Computation- and Data-intensive Particle Physics Analyses“. en. PhD thesis. 2022.
DOI: 10.5445/IR/1000143165.

[91] G. Bohm and G. Zech. „Statistics of weighted Poisson events and its applications“. *Nucl. Instrum. Meth. A* 748 (2014), pp. 1–6.
DOI: 10.1016/j.nima.2014.02.021. arXiv: 1309.1287 [physics.data-an].

[92] CMS Collaboration. „Precision luminosity measurement in proton-proton collisions at $\sqrt{s} = 13$ TeV in 2015 and 2016 at CMS“. *Eur. Phys. J. C* 81.9 (2021), p. 800.
DOI: 10.1140/epjc/s10052-021-09538-2. arXiv: 2104.01927 [hep-ex].

[93] CMS Collaboration. „CMS luminosity measurement for the 2017 data-taking period at $\sqrt{s} = 13$ TeV“. Tech. rep. Geneva: CERN, 2018.

[94] CMS Collaboration. „CMS luminosity measurement for the 2018 data-taking period at $\sqrt{s} = 13$ TeV“. Tech. rep. Geneva: CERN, 2019.

[95] M. French et al. „Design and results from the APV25, a deep sub-micron CMOS front-end chip for the CMS tracker“. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 466.2 (July 2001), pp. 359–365.
DOI: 10.1016/s0168-9002(01)00589-7.

[96] „CMS Silicon Strip Performance Results 2016“.
URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/StripsOfflinePlots2016 (visited on 07/28/2023).

[97] 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“. *Journal of High Energy Physics* 2014.7 (July 2014).
DOI: 10.1007/jhep07(2014)079.

[98] P. Artoisenet et al. „Automatic spin-entangled decays of heavy resonances in Monte Carlo simulations“. *Journal of High Energy Physics* 2013.3 (Mar. 2013).
DOI: 10.1007/jhep03(2013)015.

[99] R. Frederix and S. Frixione. „Merging meets matching in MC@NLO“. *Journal of High Energy Physics* 2012.12 (Dec. 2012).
DOI: 10.1007/jhep12(2012)061.

[100] K. Melnikov and F. Petriello. „Electroweak gauge boson production at hadron colliders through $\mathcal{O}(\alpha_s^2)$“. *Physical Review D* 74.11 (Dec. 2006), p. 114017.
DOI: 10.1103/physrevd.74.114017.

[101] R. Gavin et al. „FEWZ 2.0: A code for hadronic Z production at next-to-next-to-leading order“. *Computer Physics Communications* 182.11 (Nov. 2011), pp. 2388–2403.
DOI: 10.1016/j.cpc.2011.06.008.

[102] R. Gavin et al. „W physics at the LHC with FEWZ 2.1“. 2012.
DOI: 10.48550/ARXIV.1201.5896.

[103] Y. Li and F. Petriello. „Combining QCD and electroweak corrections to dilepton production in the framework of the FEWZ simulation code“. *Physical Review D* 86.9 (Nov. 2012), p. 094034.
DOI: 10.1103/physrevd.86.094034.

[104] T. Gehrmann et al. „W^+W^- Production at Hadron Colliders in Next to Next to Leading Order QCD“. *Phys. Rev. Lett.* 113 (21 Nov. 2014), p. 212001.
DOI: 10.1103/PhysRevLett.113.212001.

[105] S. Frixione, P. Nason, and C. Oleari. „Matching NLO QCD computations with parton shower simulations: the POWHEG method“. *Journal of High Energy Physics* 2007.11 (Nov. 2007), pp. 070–070.
DOI: 10.1088/1126-6708/2007/11/070.

[106] S. Alioli et al. „A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX“. *Journal of High Energy Physics* 2010.6 (June 2010).
DOI: 10.1007/jhep06(2010)043.

[107] S. Frixione, G. Ridolfi, and P. Nason. „A positive-weight next-to-leading-order Monte Carlo for heavy flavour hadroproduction“. *Journal of High Energy Physics* 2007.09 (Sept. 2007), pp. 126–126.
DOI: 10.1088/1126-6708/2007/09/126.

[108] M. Czakon and A. Mitov. „Top++: A program for the calculation of the top-pair cross-section at hadron colliders“. *Computer Physics Communications* 185.11 (Nov. 2014), pp. 2930–2938.
DOI: 10.1016/j.cpc.2014.06.021.

[109] M. Botje et al. „The PDF4LHC Working Group Interim Recommendations“. 2011.
DOI: 10.48550/ARXIV.1101.0538.

[110] A. D. Martin et al. „Uncertainties on α S in global PDF analyses and implications for predicted hadronic cross sections“. *The European Physical Journal C* 64.4 (Oct. 2009), pp. 653–680.
DOI: 10.1140/epjc/s10052-009-1164-2.

[111] G. Watt. „MSTW PDFs and impact of PDFs on cross sections at Tevatron and LHC“. *Nuclear Physics B - Proceedings Supplements* 222-224 (Jan. 2012), pp. 61–80.
DOI: 10.1016/j.nuclphysbps.2012.03.008.

[112] R. D. Ball et al. „Parton distributions with LHC data“. *Nuclear Physics B* 867.2 (Feb. 2013), pp. 244–289.
DOI: 10.1016/j.nuclphysb.2012.10.003.

[113] E. Re. „Single-top Wt-channel production matched with parton showers using the POWHEG method“. *The European Physical Journal C* 71.2 (Feb. 2011).
DOI: 10.1140/epjc/s10052-011-1547-z.

[114] S. Alioli et al. „NLO single-top production matched with shower in POWHEG: s- and t-channel contributions“. *Journal of High Energy Physics* 2009.09 (Sept. 2009), pp. 111–111.
DOI: 10.1088/1126-6708/2009/09/111.

[115] S. Alioli et al. „Erratum: NLO single-top production matched with shower in POWHEG: s- and t-channel contributions“. *Journal of High Energy Physics* 2010.2 (Feb. 2010).
DOI: 10.1007/jhep02(2010)011.

[116] N. Kidonakis. „Two-loop soft anomalous dimensions for single top quark associated production with a W- or H-“. *Physical Review D* 82.5 (Sept. 2010), p. 054018.
DOI: 10.1103/physrevd.82.054018.

[117] N. Kidonakis. „Top Quark Production.“ en. *Proc. of 2013* HQ2013 (2014), 139–168, DESY.
DOI: 10.3204/DESY-PROC-2013-03/KIDONAKIS.

[118] J. Campbell, T. Neumann, and Z. Sullivan. „Single-top-quark production in the t-channel at NNLO“. *JHEP* 02 (2021), p. 040.
DOI: 10.1007/JHEP02(2021)040. arXiv: 2012.01574 [hep-ph].

[119] PDF4LHC Working Group Collaboration. „The PDF4LHC21 combination of global PDF fits for the LHC Run III“. *J. Phys. G* 49.8 (2022), p. 080501.
DOI: 10.1088/1361-6471/ac7216. arXiv: 2203.05506 [hep-ph].

[120] A. G.-D. Ridder et al. „The NNLO QCD corrections to Z boson production at large transverse momentum“. 2016.
DOI: 10.48550/ARXIV.1605.04295.

[121] J. Currie et al. „Jet cross sections at the LHC with NNLOJET“. *PoS* LL2018 (2018), p. 001.
DOI: 10.22323/1.303.0001. arXiv: 1807.06057 [hep-ph].

[122] D. V. Hinkley. „On the ratio of two correlated normal random variables“. *Biometrika* 56.3 (1969), pp. 635–639.
DOI: 10.1093/biomet/56.3.635.

[123] G. Marsaglia. „Ratios of Normal Variables“. *Journal of Statistical Software* 16.4 (2006).
DOI: 10.18637/jss.v016.i04.

[124] E. Daz-Francés and F. J. Rubio. „On the existence of a normal approximation to the distribution of the ratio of two independent normal random variables“. *Statistical Papers* 54.2 (Jan. 2012), pp. 309–323.
DOI: 10.1007/s00362-012-0429-2.

[125] C. Bierlich et al. „Robust Independent Validation of Experiment and Theory: Rivet version 3“. *SciPost Phys.* 8 (2020), p. 026.
DOI: 10.21468/SciPostPhys.8.2.026. arXiv: 1912.05451 [hep-ph].

[126] J. A. Nelder and R. Mead. „A simplex method for function minimization“. English. *Comput. J.* 7 (1965), pp. 308–313. ISSN: 0010-4620.
DOI: 10.1093/comjnl/7.4.308.

[127] F. Gao and L. Han. „Implementing the Nelder-Mead simplex algorithm with adaptive parameters“. *Computational Optimization and Applications* 51.1 (May 2010), pp. 259–277.
DOI: 10.1007/s10589-010-9329-3.

[128] A. Conn, N. Gould, and P. Toint. „Trust Region Methods“. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2000. ISBN: 9780898719857.

[129] CMS Collaboration. „Minimum Bias and UE measurements at CMS“. *PoS* DIS2018 (2018), p. 036.
DOI: 10.22323/1.316.0036.

[130] G. D'Agostini. „Improved iterative Bayesian unfolding“. 2010.
DOI: 10.48550/ARXIV.1010.0632.

[131] S. Schmitt. „TUnfold, an algorithm for correcting migration effects in high energy physics“. *Journal of Instrumentation* 7.10 (Oct. 2012), T10003.
DOI: 10.1088/1748-0221/7/10/T10003.

[132] A. N. Tikhonov. „Solution of incorrectly formulated problems and the regularization method“. *Soviet Math. Dokl.* 4 (1963), pp. 1035–1038.

[133] A. Höcker and V. Kartvelishvili. „SVD approach to data unfolding“. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 372.3 (1996), pp. 469–481. ISSN: 0168-9002.
DOI: https://doi.org/10.1016/0168-9002(95)01478-0.

[134] D. A. Belsley. „Regression diagnostics. identifying influential data and sources of collinearity“. Wiley, 1980, pp. 100–104. ISBN: 0471058564.

[135] CMS Collaboration. „Luminosity recommendations for Run 2 analyses“.
URL: https://twiki.cern.ch/twiki/bin/view/CMS/LumiRecommendationsRun2 (visited on 09/14/2023).

[136] J. Alwall et al. „Comparative study of various algorithms for the merging of parton showers and matrix elements in hadronic collisions“. *Eur. Phys. J. C* 53 (2008), pp. 473–500.
DOI: 10.1140/epjc/s10052-007-0490-5. arXiv: 0706.2569 [hep-ph].

[137] CMS Collaboration. „CMS Phase-2 Computing Model: Update Document“. Tech. rep. Geneva: CERN, 2022.

[138] CMS Collaboration. „The Phase-2 Upgrade of the CMS Data Acquisition and High Level Trigger“. Tech. rep. Geneva: CERN, Mar. 2021.

[139] I. Foster and C. Kesselman, eds. „The Grid: Blueprint for a New Computing Infrastructure“. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, p. 572. ISBN: 9781558604759.

[140] K. Bos et al. „LHC computing Grid: Technical Design Report. Version 1.06 (20 Jun 2005)“. Technical design report. LCG. Geneva: CERN, 2005.

[141] I. Bird et al. „Update of the Computing Models of the WLCG and the LHC Experiments“. Tech. rep. Apr. 2014.

[142] M. Aderholz et al. „Models of Networked Analysis at Regional Centres for LHC Experiments (MONARC). Phase 2 Report.“ Tech. rep. 2000, p. 43.

[143] I. C. Legrand and H. B. Newman. „The MONARC Toolset for Simulating Large Network-Distributed Processing Systems“. *Proceedings of the 32nd Conference on Winter Simulation*. WSC '00. Orlando, Florida: Society for Computer Simulation International, 2000, pp. 1794–1801. ISBN: 0780365828.

[144] P. Malzacher et al. „Requirements for a Regional Data and Computing Centre in Germany (RDCCG)“. 2001.
URL: https://www.scc.kit.edu/downloads/SDM/GridKa/RDCCG-answer-v8.pdf (visited on 08/08/2022).

[145] H. Marten, K. Mickel, and R. Kupsch. „A Grid Computing Centre at Forschungszentrum Karlsruhe Response on the Requirements for a Regional Data and Computing Centre in Germany (RDCCG)“. 2001.
URL: https://www.scc.kit.edu/downloads/SDM/GridKa/RDCCG-answer-v8.pdf (visited on 08/08/2022).

[146] „WLCG Monitoring & Visualisation“. 2022.
URL: https://wlcg.web.cern.ch/using-wlcg/monitoring-visualisation (visited on 06/12/2023).

[147] Docker, Inc. „Docker Website“. 2013.
URL: https://www.docker.com/ (visited on 08/08/2022).

[148] LF Projects, LLC. „Apptainer“. 2021.
URL: https://apptainer.org/ (visited on 08/08/2022).

[149] G. M. Kurtzer, V. Sochat, and M. W. Bauer. „Singularity: Scientific containers for mobility of compute“. *PLOS ONE* 12.5 (May 2017), pp. 1–20.
DOI: 10.1371/journal.pone.0177459.

[150] J. Blomer et al. „The CernVM File System: v2.7.5“. en. 2020.
DOI: 10.5281/ZENODO.1010441.

[151] Amazon.com, Inc. „Amazon Web Services“. 2002.
URL: https://aws.amazon.com/ (visited on 06/06/2023).

[152] Google. „Google Cloud Platform“. 2008.
URL: https://cloud.google.com/ (visited on 06/06/2023).

[153] Microsoft Corporation. „Microsoft Azure“. 2008.
URL: https://azure.microsoft.com/ (visited on 06/06/2023).

[154] F. B. Megino et al. „Seamless integration of commercial Clouds with ATLAS Distributed Computing“. *EPJ Web of Conferences* 251 (2021). Ed. by C. Biscarat et al., p. 02005.
DOI: 10.1051/epjconf/202125102005.

[155] E. Martelli et al. „LHCONE - Large Hadron Collider Open Network Environment“.
URL: https://twiki.cern.ch/twiki/bin/view/LHCONE/WebHome?rev=86.

[156] B. Hoeft, E. Martelli, et al. „LHCONE Acceptable Use Policy“.
URL: https://twiki.cern.ch/twiki/bin/view/LHCONE/LhcOneAup (visited on 08/14/2023).

[157] I. Sfiligoi. „glideinWMSa generic pilot-based workload management system“. *Journal of Physics: Conference Series* 119.6 (July 2008), p. 062044.
DOI: 10.1088/1742-6596/119/6/062044.

[158] I. Sfiligoi et al. „The Pilot Way to Grid Resources Using glideinWMS“. *2009 WRI World Congress on Computer Science and Information Engineering*. IEEE, 2009.
DOI: 10.1109/csie.2009.950.

[159] I. Sfiligoi et al. „CMS experience of running glideinWMS in High Availability mode“. *Journal of Physics: Conference Series* 513.3 (June 2014), p. 032086.
DOI: 10.1088/1742-6596/513/3/032086.

[160] S. Belforte et al. „Evolution of the pilot infrastructure of CMS: towards a single glideinWMS pool“. *Journal of Physics: Conference Series* 513.3 (June 2014), p. 032041.
DOI: 10.1088/1742-6596/513/3/032041.

[161] F. B. Megino et al. „ATLAS WORLD-cloud and networking in PanDA“. *Journal of Physics: Conference Series* 898 (Oct. 2017), p. 052011.
DOI: 10.1088/1742-6596/898/5/052011.

[162] F. B. Megino et al. „ATLAS Global Shares implementation in PanDA“. *EPJ Web of Conferences* 214 (2019). Ed. by A. Forti et al., p. 03025.
DOI: 10.1051/epjconf/201921403025.

[163] HTCondor Team. „HTCondor“. en. 2022.
DOI: 10.5281/ZENODO.2579447.

[164] A. B. Yoo, M. A. Jette, and M. Grondona. „SLURM: Simple Linux Utility for Resource Management“. *Job Scheduling Strategies for Parallel Processing*. Springer Berlin Heidelberg, 2003, pp. 44–60.
DOI: 10.1007/10968987_3.

[165] Free Software Foundation. „Slurm Workload Manager“. 2022.
URL: https://slurm.schedmd.com/ (visited on 08/22/2022).

[166] W. Allcock et al. „The Globus Striped GridFTP Framework and Server“. *ACM/IEEE SC 2005 Conference (SC'05)*. IEEE.
DOI: 10.1109/sc.2005.72.

[167] D. Alvise et al. „XRootD- A highly scalable architecture for data access“. Apr. 2005.
DOI: 10.1.1.127.9281.

[168] CMS Collaboration. „CMS XRootD Architecture and AAA“. 2010.
URL: https://twiki.cern.ch/twiki/bin/view/CMSPublic/CMSXrootDArchitecture (visited on 06/06/2023).

[169] M. Barisits et al. „Rucio: Scientific Data Management“. *Computing and Software for Big Science* 3.1 (Aug. 2019).
DOI: 10.1007/s41781-019-0026-3.

[170] E. Vaandering. „Transitioning CMS to Rucio Data Managment“. *EPJ Web of Conferences* 245 (2020). Ed. by C. Doglioni et al., p. 04033.
DOI: 10.1051/epjconf/202024504033.

[171] M. Hoseinzadeh. „A Survey on Tiering and Caching in High-Performance Storage Systems“. 2019.
DOI: 10.48550/ARXIV.1904.11560.

[172] M. J. Schnepf et al. „Dynamic Integration and Management of Opportunistic Resources for HEP“. *EPJ Web of Conferences* 214 (2019). Ed. by A. Forti et al., p. 08009.
DOI: 10.1051/epjconf/201921408009.

[173] G. Erli et al. „roced-scheduler/ROCED 1.1.0“. 2018.
DOI: 10.5281/ZENODO.1888234.

[174] M. Fischer et al. „Lightweight dynamic integration of opportunistic resources“. *EPJ Web of Conferences* 245 (2020). Ed. by C. Doglioni et al., p. 07040.
DOI: 10.1051/epjconf/202024507040.

[175] M. Fischer et al. „MatterMiners/cobald: v0.12.3“. 2021.
DOI: 10.5281/ZENODO.1887872.

[176] M. Giffels et al. „MatterMiners/tardis: The Survivors“. 2021.
DOI: 10.5281/ZENODO.2240605.

[177] F. Berghaus et al. „High-Throughput Cloud Computing with the Cloudscheduler VM Provisioning Service“. *Computing and Software for Big Science* 4.1 (Feb. 2020).
DOI: 10.1007/s41781-020-0036-1.

[178] B. Holzman et al. „HEPCloud, a New Paradigm for HEP Facilities: CMS Amazon Web Services Investigation“. *Computing and Software for Big Science* 1.1 (Sept. 2017).
DOI: 10.1007/s41781-017-0001-9.

[179] R. F. von Cube et al. „Opportunistic transparent extension of a WLCG Tier 2 center using HPC resources“. *EPJ Web of Conferences* 251 (2021). Ed. by C. Biscarat et al., p. 02059.
DOI: 10.1051/epjconf/202125102059.

[180] M. Fischer et al. „Effective Dynamic Integration and Utilization of Heterogenous Compute Resources“. *EPJ Web of Conferences* 245 (2020). Ed. by C. Doglioni et al., p. 07038.
DOI: 10.1051/epjconf/202024507038.

[181] M. Böhler et al. „Transparent Integration of Opportunistic Resources into the WLCG Compute Infrastructure“. *EPJ Web of Conferences* 251 (2021). Ed. by C. Biscarat et al., p. 02039.
DOI: 10.1051/epjconf/202125102039.

[182] K. Fransham et al. „Research computing in a distributed cloud environment“. *Journal of Physics: Conference Series* 256 (Nov. 2010), p. 012003.
DOI: 10.1088/1742-6596/256/1/012003.

[183] R. Sobie. „Utilizing clouds for Belle II“. *Journal of Physics: Conference Series* 664.2 (Dec. 2015), p. 022037.
DOI: 10.1088/1742-6596/664/2/022037.

[184] R. Seuster et al. „Context-aware distributed cloud computing using CloudScheduler“. *Journal of Physics: Conference Series* 898 (Oct. 2017), p. 052039.
DOI: 10.1088/1742-6596/898/5/052039.

[185] C. D. Hauck, M. Herty, and G. Visconti. „Qualitative Properties of Mathematical Model For Data Flow“. 2020. arXiv: 1910.10117 [math.AP].

[186] S. Bagchi. „The Modeling Approaches of Distributed Computing Systems“. *Software Engineering, Business Continuity, and Education*. Ed. by T.-h. Kim et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 479–488. ISBN: 978-3-642-27207-3.

[187] P. Velho et al. „On the Validity of Flow-Level Tcp Network Models for Grid and Cloud Simulations“. *ACM Trans. Model. Comput. Simul.* 23.4 (Dec. 2013). ISSN: 1049-3301.
DOI: 10.1145/2517448.

[188] R. C. Barnard, K. Huang, and C. Hauck. „A mathematical model of asynchronous data flow in parallel computers“. 2019. arXiv: 1910.09305 [cs.DC].

[189] A. Kostin and L. Ilushechkina. „Modeling and simulation of distributed systems : with CD-ROM“. World Scientific, 2010. ISBN: 9814291676; 9789814291675.

[190] F. Bause and P. S. Kritzinger. „Stochastic petri nets“. Vol. 1. Vieweg Wiesbaden, 2002. ISBN: 3-528-15535-3.

[191] H. Hermanns, U. Herzog, and J.-P. Katoen. „Process algebra for performance evaluation“. *Theoretical Computer Science* 274.1 (2002). Ninth International Conference on Concurrency Theory 1998, pp. 43–87. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(00)00305-4.

[192] J. Cao et al. „Performance modeling of parallel and distributed computing using PACE“. *Conference Proceedings of the 2000 IEEE International Performance, Computing, and Communications Conference (Cat. No.00CH37086)*. 2000, pp. 485–492. DOI: 10.1109/PCCC.2000.830354.

[193] V. Grassi, R. Mirandola, and A. Sabetta. „Filling the gap between design and performance/reliability models of component-based systems: A model-driven approach“. *Journal of Systems and Software* 80.4 (Apr. 2007), pp. 528–558. DOI: 10.1016/j.jss.2006.07.023.

[194] D. Hamlet, D. Mason, and D. Woit. „Component-Based Software Development: Case Studies“. Ed. by K.-K. Lau. Vol. 1. Chapter: Properties of Software Systems Synthesized from Components. World Scientific Publishing Company, 2004. DOI: 10.1142/5526.

[195] E. Eskenazi, A. Fioukov, and D. Hammer. „Performance prediction for component compositions“. *Component-Based Software Engineering: 7th International Symposium, CBSE 2004, Edinburgh, UK, May 24-25, 2004. Proceedings 7*. Springer. 2004, pp. 280–293.

[196] G. F. Riley and T. R. Henderson. „The ns-3 Network Simulator“. *Modeling and Tools for Network Simulation*. Ed. by K. Wehrle, M. Güne, and J. Gross. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34. ISBN: 978-3-642-12331-3. DOI: 10.1007/978-3-642-12331-3_2.

[197] K. Fujiwara and H. Casanova. „Speed and Accuracy of Network Simulation in the SimGrid Framework“. *Proceedings of the 2nd International ICST Conference on Performance Evaluation Methodologies and Tools*. ICST, 2007. DOI: 10.4108/nstools.2007.2010.

[198] L. Bobelin et al. „Scalable Multi-Purpose Network Representation for Large Scale Distributed System Simulation“. *CCGrid 2012 – The 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. Ottawa, Canada, May 2012, p. 19.

[199] C. Dobre and C. Stratan. „MONARC Simulation Framework“ (June 2011). arXiv: 1106.5158 [cs.DC].

[200] K. Stockinger et al. „OptorSim: a Simulation Tool for Scheduling and Replica Optimisation in Data Grids“. en. 2005.
DOI: 10.5170/CERN-2005-002.707.

[201] S. Ostermann, R. Prodan, and T. Fahringer. „Dynamic Cloud provisioning for scientific Grid workflows“. *2010 11th IEEE/ACM International Conference on Grid Computing*. 2010, pp. 97–104.
DOI: 10.1109/GRID.2010.5697953.

[202] R. Buyya and M. Murshed. „GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing“. *Concurrency and Computation: Practice and Experience* 14.13-15 (Nov. 2002), pp. 1175–1220.
DOI: 10.1002/cpe.710.

[203] R. N. Calheiros et al. „CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms“. *Software: Practice and Experience* 41.1 (Aug. 2010), pp. 23–50.
DOI: 10.1002/spe.995.

[204] H. Casanova et al. „Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms“. *Journal of Parallel and Distributed Computing* 74.10 (Oct. 2014), pp. 2899–2917.
DOI: 10.1016/j.jpdc.2014.06.008.

[205] CLOUDS Laboratory, University of Melbourne. „PUBLICATIONS: Dr. Buyya with his team and colleagues“. [Accessed 15-Jun-2023]. 2020.
URL: http://www.cloudbus.org/publications-years.html (visited on 06/15/2023).

[206] SimGrid Team. „They use SimGrid — simgrid.org“. [Accessed 15-Jun-2023].
URL: https://simgrid.org/usages.html (visited on 06/15/2023).

[207] M. Horzela et al. „HEPCompSim/DCSim: DCSim simulator release v0.3“. 2023.
DOI: 10.5281/ZENODO.8300961.

[208] M. Horzela et al. „Modelling Distributed Heterogeneous Computing Infrastructures for HEP Applications“. 26th International Conference on Computing in High Energy and Nuclear Physics. 2023.

[209] H. Casanova et al. „WRENCH: A Framework for Simulating Workflow Management Systems“. *2018 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS)*. 2018, pp. 74–85.
DOI: 10.1109/WORKS.2018.00013.

[210] H. Casanova et al. „Developing Accurate and Scalable Simulators of Production Workflow Management Systems with WRENCH“. *Future Generation Computer Systems* 112 (2020), pp. 162–175.
DOI: 10.1016/j.future.2020.05.030.

[211] D. P. Bertsekas. „Data networks“. Prentice Hall, 1992, pp. 328, 524–529. ISBN: 0132009161.

[212] D. M. Chiu. „Some observations on fairness of bandwidth sharing“. *Proceedings ISCC 2000. Fifth IEEE Symposium on Computers and Communications*. 2000, pp. 125–131.
DOI: 10.1109/ISCC.2000.860626.

[213] G. Marfia et al. „TCP Libra: Exploring RTT-Fairness for TCP“. *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*. Springer Berlin Heidelberg, 2007, pp. 1005–1013.
DOI: 10.1007/978-3-540-72606-7_86.

[214] M. Heusse et al. „Two-way TCP connections“. *ACM SIGCOMM Computer Communication Review* 41.2 (Apr. 2011), pp. 5–15.
DOI: 10.1145/1971162.1971164.

[215] New Mexico State University. „Message Parsing Interface“.
URL: https://hpc.nmsu.edu/discovery/mpi/introduction/ (visited on 06/22/2023).

[216] WRENCH Team. „WRENCH release v2.2“.
URL: https://github.com/wrench-project/wrench/releases/tag/v2.2 (visited on 10/10/2023).

[217] CMS Collaboration. „The Higgs Boson turns 10: Results from the CMS experiment“.
URL: https://cms.cern/news/higgs-boson-turns-10-results-cms-experiment.

[218] CMS Collaboration. „Measurements of Higgs boson production in the decay channel with a pair of τleptons in proton-proton collisions at $\sqrt{s} = 13$TeV“. *The European Physical Journal C* 83.7 (2023), p. 562. ISSN: 1434-6052.
DOI: 10.1140/epjc/s10052-023-11452-8.

[219] C. Heidecker. „Jet Momentum Resolution for the CMS Experiment and Distributed Data Caching Strategies“. PhD thesis. Karlsruhe Institute of Technology (KIT), 2020.

[220] A. Rizzi, G. Petrucciani, and M. Peruzzi. „A further reduction in CMS event data for analysis: the NANOAOD format“. *EPJ Web of Conferences* 214 (2019). Ed. by A. Forti et al., p. 06021.
DOI: 10.1051/epjconf/201921406021.

[221] G. Petrucciani, A. Rizzi, and C. Vuosalo. „Mini-AOD: A New Analysis Data Format for CMS“. *Journal of Physics: Conference Series* 664.7 (Dec. 2015), p. 072052.
DOI: 10.1088/1742-6596/664/7/072052.

[222] M. Peruzzi, G. Petrucciani, and A. Rizzi. „The NanoAOD event data format in CMS“. *Journal of Physics: Conference Series* 1525.1 (Apr. 2020), p. 012038.
DOI: 10.1088/1742-6596/1525/1/012038.

[223] WLCG Collaboration. „HEPiX Benchmark Working group“. 2022.
URL: https://w3.hepix.org/benchmarking.html (visited on 07/19/2023).

[224] J. L. Henning. „SPEC CPU2006 Benchmark Descriptions“. *SIGARCH Comput. Archit. News* 34.4 (Sept. 2006), pp. 1–17. ISSN: 0163-5964. DOI: 10.1145/1186736.1186737.

[225] Komitee für Elementarteilchenphysik. „Perspektivpapier der Teilchenphysiker:innen in Deutschland“. URL: https://www.ketweb.de/sites/site_ketweb/content/e199639/e312771/KET-Computing-Strategie-HL-LHC-final.pdf.

Danksagung

Zu guter Letzt möchte ich die Gelegenheit nutzen, um Danke zu sagen. Oder in gebrochenem Quenya:

(hantanyë tyen!)

Danke an meine Referenten Günter Quast und Achim Streit, ohne deren Förderung diese Arbeit nie Zustande gekommen wäre. Vielen Dank für die Chance die Promotion zu starten, die Herausforderungen an denen ich wachsen konnte, die fachliche Hilfe und seelische Unterstützung bei der Bewältigung dieser Herausforderungen, und die Lösung der bürokratischen Hürden. Dies hat mir die erfolgreiche Forschung ermöglicht, die in dieser Arbeit dokumentiert ist.

Besonderer Dank gebührt meinen Kollegen aus aller Welt und Fachrichtungen, die mir mit Rat und Tat zur Seite standen. Insbesondere möchte ich mich bei Henri, Nils, Stefan, Artur, Robin, Simone, Fred, Cedric und vielen mehr für ihre Hilfe, Inspiration und ihren Zuspruch bedanken. Ohne euch wäre die akademische Reise ein einsames Unterfangen geworden.

Besonderer Dank gebührt ebenfalls meiner Familie und meinen Freunden, die mit uneingeschränktem Beistand und bedingungsloser Unterstützung meine bisweilen körperliche und geistige Abwesenheit in dieser Zeit ertragen haben. Ohne euch und dem von euch gebotenen sicheren Rückhalt hätte ich es nicht so weit geschafft.

Zu guter Letzt Danke an diejenigen, die sich die Mühe gemacht haben diese Thesis bis hierhin zu lesen. Meine Anerkennung! Ich hoffe, ich habe euch nicht gelangweilt, und ihr konntet auch ein wenig persönlich von meiner Arbeit profitieren. Durch euch wird dieses Dokument zu mehr als nur einem formellen Kriterium.