Jet Substructure Modeling in Dijet and Z+Jet Events

Julian Zeyn

University of Hamburg

October 2020

Supervisor: Dr. Andreas Hinzmann

Second Supervisor: Prof. Dr. Peter Schleper

Abstract

Searches for boosted W-, Z-, H- or top-jets make use of jet substructure observables to reduce the background composed of quark- and gluon-jets. In particular, gluon-jets dominate the QCD multi-jet background. Therefore, precise background modelling is necessary to gain sensitivity for such searches. Gluon enriched dijet and quark enriched Z+jet event samples were measured to examine the simulation of substructure observables. In this thesis, the modelling of quark and gluon jets in Monte Carlo generators is studied, with the goal of finding a set of simulation parameters (tune) that well describes the jet substructure of quark- and gluon-jets while not breaking the modelling of existing underlying event measurements. In particular, the effects of varying α_S on the substructure simulation was studied for Pythia8 and Herwig7. Additionally, for Pythia8 the Colour Reconnection models were examined. The description of quark-jets was better than gluon-jets, but there was no perfect description for all substructure variables achieved by any of the configurations under study. Compared to Pythia8, Herwig7 shows a more promising ability to describe jet substructure and underlying events simultaneously.

Zusammenfassung

Suchen nach geboosteten W-, Z-, H-, oder Top-Jets nutzen Jetsubstruktur-Observablen um den Untergrund von Quark- und Gluon-Jets zu reduzieren. Insbesondere Gluon-Jets dominieren den QCD-Multijetuntergrund. Daher ist eine präzise Modellierung des Untergrunds notwendig, um die Sensitivität für solche Suchen zu erhöhen. Mit Gluon-Jets angereicherte Dijet- und mit Quark-Jets angereicherte Z+Jet-Events wurden gemessen, um die Simulation von Substrukturobservablen zu untersuchen. In dieser Masterarbeit wurde die Modellierung von Quark- und Gluon-Jets in Monte Carlo-Generatoren analysiert, mit dem Ziel, eine Parametrisierung zu finden, die die Jetsubstruktur von Quark- und Gluon-Jets gut beschreibt, ohne dabei die Modellierung von bereits existierenden Underlying Event-Messungen zu verschlechtern. Dafür wurde der Einfluss von α_S auf die Jetsubstruktur-Simulation für Pythia8 und Herwig7 untersucht. Außerdem wurden die Colour Reconnection-Modelle von Pythia8 studiert. Eine bessere Beschreibung von Quark-Jets als von Gluon-Jets wurde beobachtet, aber es wurde von keiner untersuchten Konfiguration eine perfekte Beschreibung von allen Substrukturvariablen erreicht. Verglichen mit Pythia8 zeigt Herwig7 eine vielversprechende Fähigkeit, Jetsubstruktur und Underlying Event gleichzeitig zu beschreiben.

Contents

Chapter 1.

Introduction

Many particles produced in interactions in proton-proton collisions are partons (quarks and gluons). These partons split into more partons via the strong interaction before then hadronizing. This is called a parton shower. The hadrons are detected and formed into a jet by an algorithm. These jets are used as an representation of the initial parton for the reconstruction of the proton-proton collision. Additionally, the jet substructure, which describes the number of partons and their distribution in a jet, can be evaluated to differentiate between a jet originating from a quark or gluon, called quark- or gluon-jet, respectively [1].
To improve this evaluation, the jet substructure needs to be well understood and simulated. It is known, that quarks and gluons have different Casimir colour factors, $C_F = \frac{4}{3}$ for quarks and $C_A = 3$ for gluons. Due to this, they have a different probability to emit a soft gluon ($C_A/C_F = \frac{9}{4}$). Therefore, gluon-jets should contain more particles on average and these particles should have a wider spread within the jet. But there are also less well understood factors effecting the jet substructure, e.g. the modelling of the hadronization and colour reconnection. These are only evaluated in Monte-Carlo (MC) generators using phenomenological models. Quark jets have already been well constrained in precious measurements, gluon jets, however, are less well understood [2].
For the data used in this thesis, a set of five jet substructure observables, the generalised angularities λ_β^κ [3], were measured in gluon-jet enriched dijet and quark-jet enriched Z+jet event samples [4].
Chapters 2 and 3 give an introduction into the theoretical and experimental fundamentals, respectively. Previous measurements used for this thesis are described in Chapter 4 and the varied simulation parameters are explained in Chapter 5. Finally, the results are presented in Chapter 6.

In this thesis, the effects of varying α_S on the substructure simulation was studied for Pythia8 and Herwig7 in Chapter 6.1. Additionally, in Chapter 6.2, the Colour Reconnection models were examined for Pythia8. In Chapter 6.3, Pythia8 and Herwig7 are compared in their ability to simulate the jet substructure observables. Finally, in Chapters 6.4 and 6.5, the modelling of quark- and gluon-jets and the effect of varying α_S on the underlying event tune, respectively, are presented.

Chapter 2.

Theoretical Introduction

In this chapter, the theoretical foundations of my thesis are discussed. In Section 2.1, the Standard Model of particle physics is presented based on [5]. Section 2.2 introduces the Monte Carlo generators used and Section 2.3 describes underlying events.

2.1. Standard Model

The Standard Model of particle physics describes all known elementary particles and their interactions. In Figure 2.1, the fermions, spin-$\frac{1}{2}$ particles from which the known matter is built, and the bosons, integer-spin particles which mediate the interactions, are shown. For each particle, its mass, charge and spin are displayed.
The fermions can be further divided into quarks and leptons. Both quarks and leptons have three generations. The first generation consists of up- and down-quark, the electron and electron neutrino; the second of charm- and strange-quark, muon and muon neutrino; and the third generation of top- and bottom-quark, tau and tau neutrino. The quarks have an electrical charge of either $+\frac{2}{3}e$, called up-type quarks, or $-\frac{1}{3}e$, the down-type quarks. Additionally, quarks carry colour charge with three different states. The colour charged particles (quarks and gluons) cannot be isolated, this is called colour confinement. Quarks and gluons instead form hadrons, which are colour neutral. Hadrons made of one quark and one antiquark are called mesons, while baryons are made of three quarks [6]. The leptons are further divided into charge $-1e$, the electron, muon and tau, and charge 0 leptons, the three respective neutrinos. All fermions also carry a weak isospin. It is $+\frac{1}{2}$ for the up-type quarks and neutrinos, and $-\frac{1}{2}$ for the down-type quarks, electron, muon and tau [7]. Particles in higher

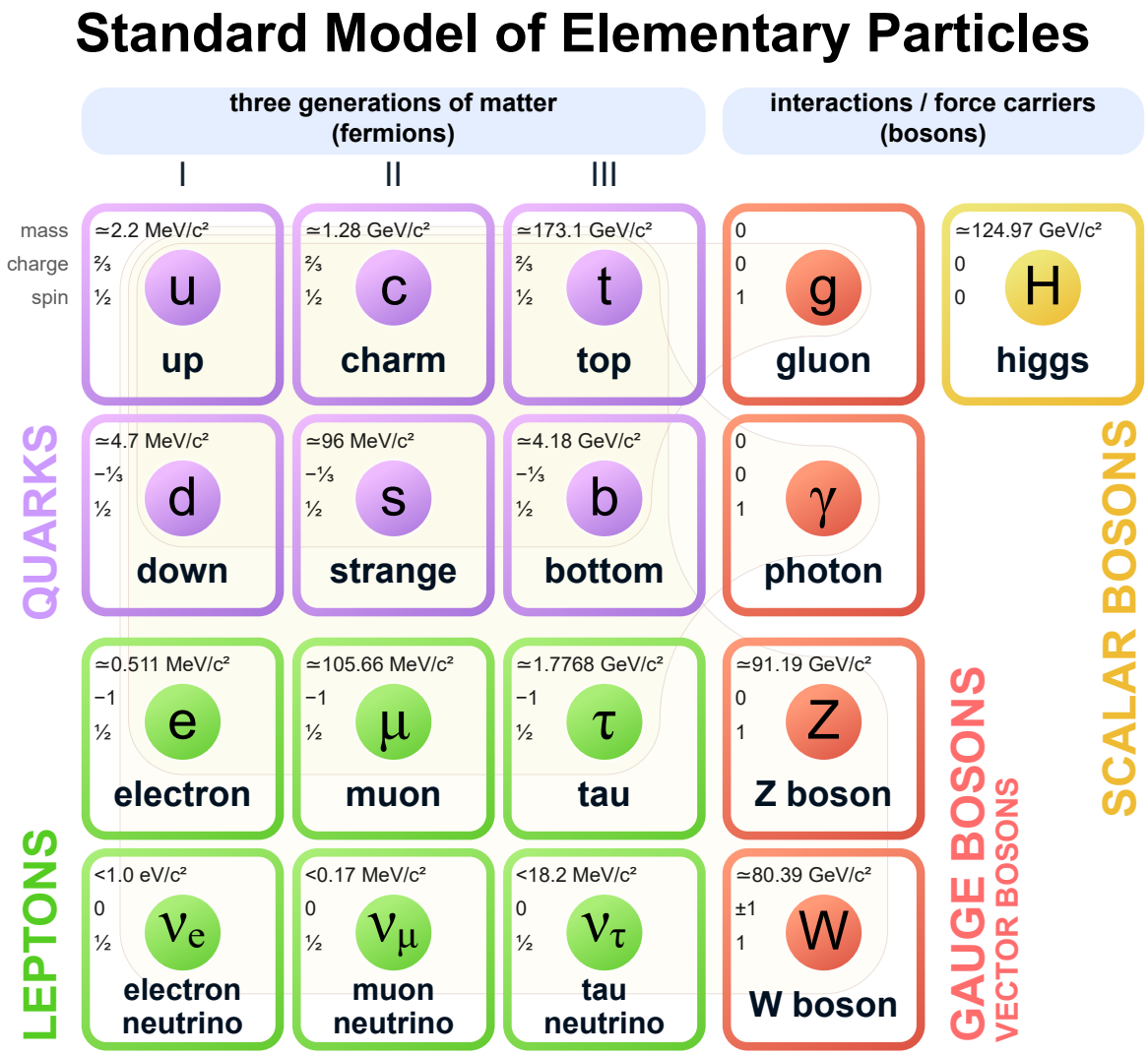


Figure 2.1.: Overview over the elementary particles of the Standard Model. In the top left corner of each particle are its mass, charge and spin displayed. For the three neutrinos, their upper mass limits are shown, although they are massless in the Standard Model. Taken from [5].

generations were discovered more recently, and have heavier masses than those in lower generations.
The bosons are the gluon, photon, Z- and $W^{\pm}$-bosons and the Higgs. The massless gluon, which is the force carrier for the strong force, couples only to quarks ad gluons as they are the only particles with colour charge. The massless photon couples to all charged particles and is the mediator of the electromagnetic interaction. The carriers of the weak force are the massive neutral Z- and charged $W^{\pm}$-bosons, these couple to the weak isospin of the fermions. The final boson is the Higgs, which can explain how the particles of the Standard Model gain their masses.
Additionally, every particle has its anti-particle with same mass but opposite sign electric and colour charge and weak isospin. For example, the anti-particle of the electron is the positron (or anti-electron), and is naturally produced in radioactive decay. Some particles, as the photon, are its own anti-particle [8].

2.2. Generators

This chapter is a short introduction to Monte Carlo (MC) generators based on [9].
As protons are composite particles, the collision of two protons is significantly more complex than the collision of leptons. In Figure 2.2, a sketch of a proton-proton collision is shown. Therefore, Monte Carlo generators are the only option to describe these events. MC generators contain theory of the physics in the collision and make use of models for less understood physics. They can describe, among others, hard and soft interactions, initial- and final-state parton showers, multiparton interactions and hadronization. MC generators usually split the interaction into subprocesses and process them one after the other. They typically start with selecting the partons for the hard scattering process using Parton Density Functions (PDF). Then, these partons can generate an initial-state shower. The next step is the simulation of the hard process (parton-parton scattering), followed by the generation of final-state parton showers. Then, the underlying events are processed. The underlying events are model that needs to be tuned and are described in Section 2.3. In the final steps, Hadrons are formed and unstable particles decay.
For this thesis, Pythia8 [10] [11] and Herwig7 [12] are used. The next two sections describe these a bit more.

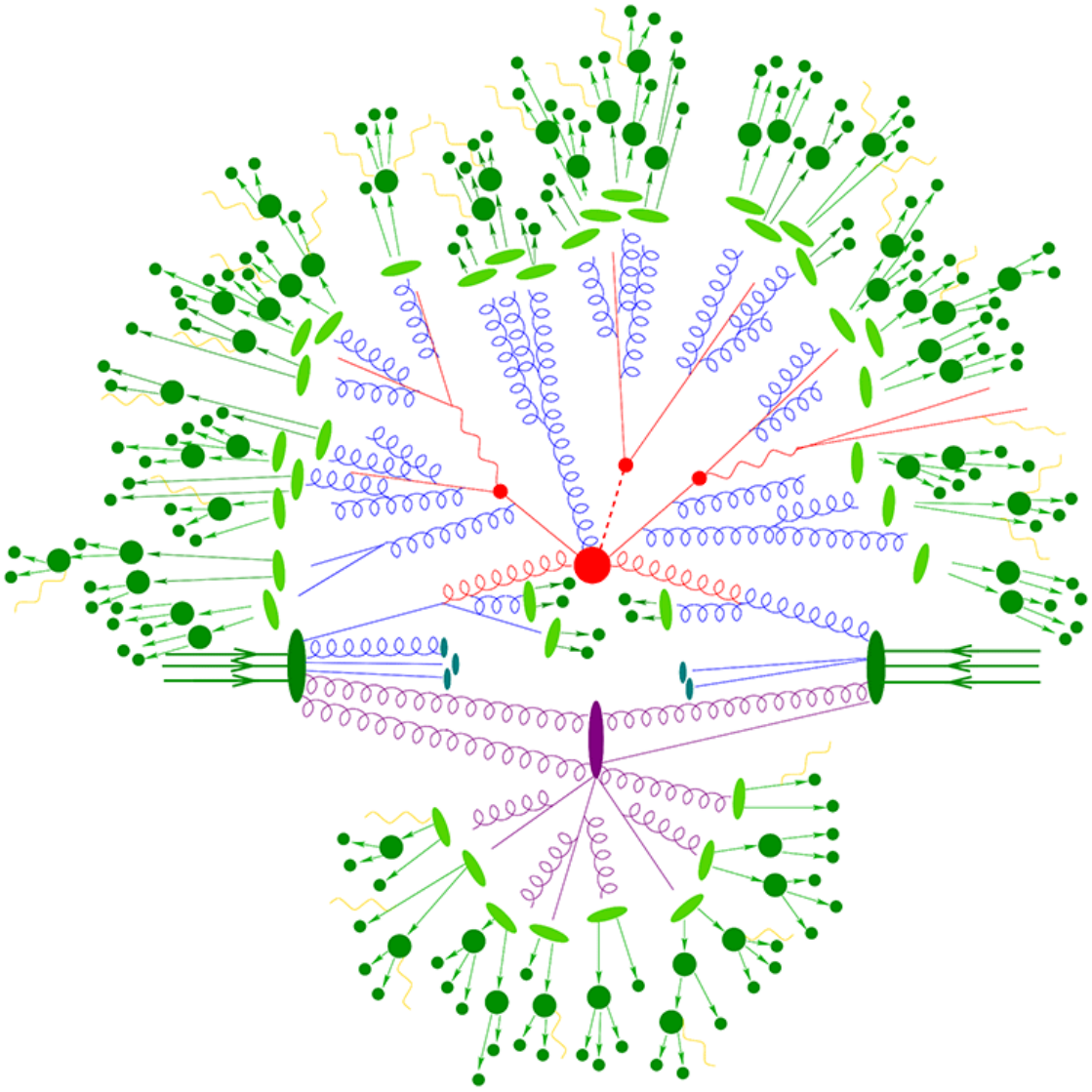


Figure 2.2.: Scheme of a proton-proton collision with the main vertex, secondary vertices and parton shower in red, a multiple-parton interaction and its parton shower in violet, hadronization in light green and dark green hadrons. Taken from [9].

2.2.1. Pythia8

Pythia8 is used to generate high energy physics events for both Standard Model and Beyond the Standard Model processes. It is designed with attention on hadronic processes where the physics are not well understood. Pythia8 therefore uses QCD-based models to describe such events. It divides the simulation into multiple steps: the two incoming particles first generate initial state radiation and then scatter into outgoing partons. The outgoing partons then generate a final state parton shower and multiple parton interactions are simulated. Then, the beam remnants and their colour states are considered before the outgoing partons are hadronized and, finally, the hadrons may decay.
For more information, please refer to [10] and [11].

2.2.2. Herwig7

Herwig7 is a MC generator for lepton-lepton, lepton-hadron and hadron-hadron collisions. It can simulate not only Standard Model processes, but also Beyond the Standard Model (BSM) processes. A process is simulated in a few steps: First, in the elementary hard process, the incoming particles collide and produce the primary outgoing particles. In the next step, the initial- and final-state parton showers are evolved until the energy of the partons is below a cut-off scale. Then, heavy elementary particles, e.g. top-quarks and the Higgs boson, decay and afterwards, multiple parton scatterings are simulated for large centre-of-mass energies. The partons below the energy cut-off scale then hadronize and finally the hadrons decay.
For more information, please refer to [12].

2.3. Underlying Events

Underlying events are defined as the particles not originating from the main parton-parton scattering. It consists of initial-state radiation, final-state radiation not originating from the main scattering, beam-beam remnants, multiple-parton interactions, pile-up and noise.
Figure 2.3 shows the division of an event into ϕ regions. The transverse regions are the most sensitive to the modelling of the UE and can be further divided into the transMin

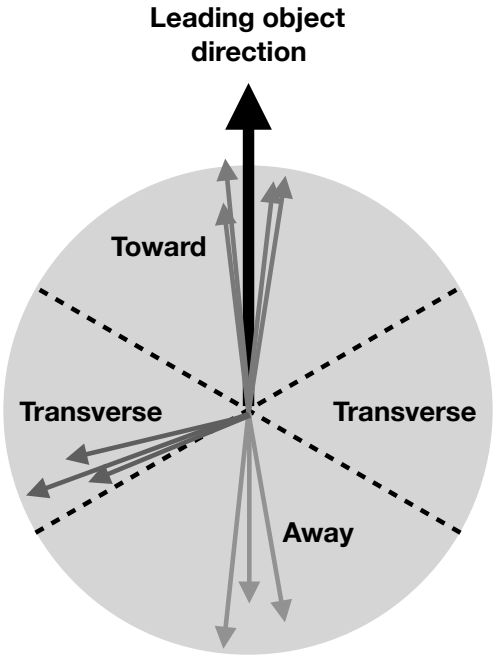


Figure 2.3.: Scheme of ϕ regions relative to the leading object. The toward region is defined as $|\Delta\phi| \leq 60^{\circ}$, the away region as $|\Delta\phi| > 120^{\circ}$ and the transverse regions as $60 < |\Delta\phi| \leq 120^{\circ}$. Taken from [13].

and transMax regions, which contain the less and more particles, respectively. These regions are used for the tuning of underlying event models in Section 4.4.
This section is based on [14] and [13].

Chapter 3.

Experimental Introduction

In this chapter, the experimental foundations of my thesis are discussed. In Section 3.1, the Large Hadron Collider (LHC) is discussed based on [15]. One of its experiments is the Compact Muon Solenoid (CMS) experiment, where the data for this thesis was taken and which is described in Section 3.2 based on [16]. In Section 3.3, an overview over the reconstruction of events is given and Section 3.4 depicts what jets are and how the anti-k_t jet clustering algorithm works.

3.1. LHC

The LHC is a hadron collider at CERN (Conseil européen pour la recherche nucléaire). It is designed to collide proton beams with a centre-of-mass energy of 14 TeV at a peak luminosity of $L = 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. It was created to, among other goals, search for the Higgs boson, which was discovered at the LHC in 2012, and to detect new physics. It was built in the already existing 26.7 km tunnel constructed for the LEP. The beams consist of proton bunches and are collided at the four experiment-points.
Figure 3.1 shows the LHC and its four detectors ALICE [18], used for heavy-ion experiments, ATLAS [19] and CMS [16], which are used in proton-proton and heavy-ion experiments, and LHCb [20], where CP violation and rare decays of B-hadrons are measured. The figure also depicts the pre-accelerators used to increase the energy of the protons to 450 GeV before they are injected to the LHC. The protons are accelerated in multiple steps starting from LINAC2 (Linear Accelerator) via the Booster, Proton Synchrotron (PS) and Super Proton Synchrotron (SPS).
In the LHC, the protons are further accelerated from 450 GeV to 7 TeV via two indepen-

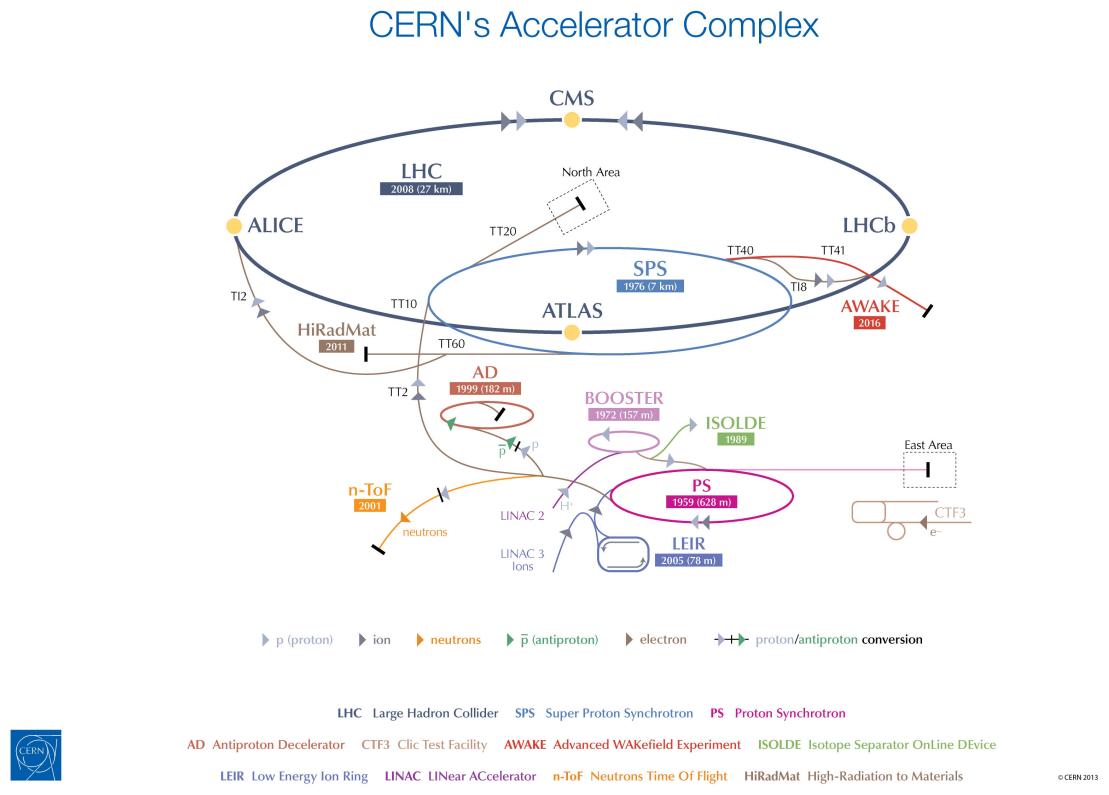


Figure 3.1.: Schematic overview of the Large Hadron Collider (LHC) and its pre-accelerators and the experiments at the CERN accelerator complex. Taken from [17].

dent superconducting radio frequency cavities. The proton beam is kept on its path using 1232 superconducting dipole magnets and focused with 392 superconducting quadrupole magnets. Further, higher order magnets are used to account for higher order effects to stabilize the beams.
The number of events per second for any given event can be calculated via

$$N_{Event} = L * \sigma_{Event} \tag{3.1}$$

where σ_{Event} is the cross section of the event and L the luminosity of the LHC. The luminosity is given by

$$L = \frac{N_b^2 n_b f_{rev} \gamma_r}{4\pi\epsilon_n \beta^*} F \tag{3.2}$$

where the symbols mean:

- N_b: number of particles per bunch
- n_b: number of bunches per beam
- f_{rev}: revolution frequency
- γ_r: relativistic gamma factor
- ϵ_n: normalized transverse beam emittance
- β^*: beta function at the collision point (corresponds to the size of the beam)
- F: geometric luminosity reduction factor due to the crossing angle at the interaction point

With $n_b = 2808$ bunches per beam, $N_b = 1.15 \times 10^{11}$ protons per bunch, $f_{rev} = 11.25\,\text{kHz}$, $\epsilon_n = 3.75\,\mu\text{m}$ and $\beta^* = 0.55\,\text{m}$, Equation 3.2 gives a luminosity of $L = 10^{34}\,\text{cm}^{-2}\,\text{s}^{-1}$.

3.2. CMS detector

The Compact Muon Solenoid (CMS) detector is located at one of the four collision points at the LHC. It is 28.7 m long, has a diameter of 15 m and weights 14 000 t. It includes five main parts: silicon trackers, crystal electromagnetic calorimeter (ECAL),

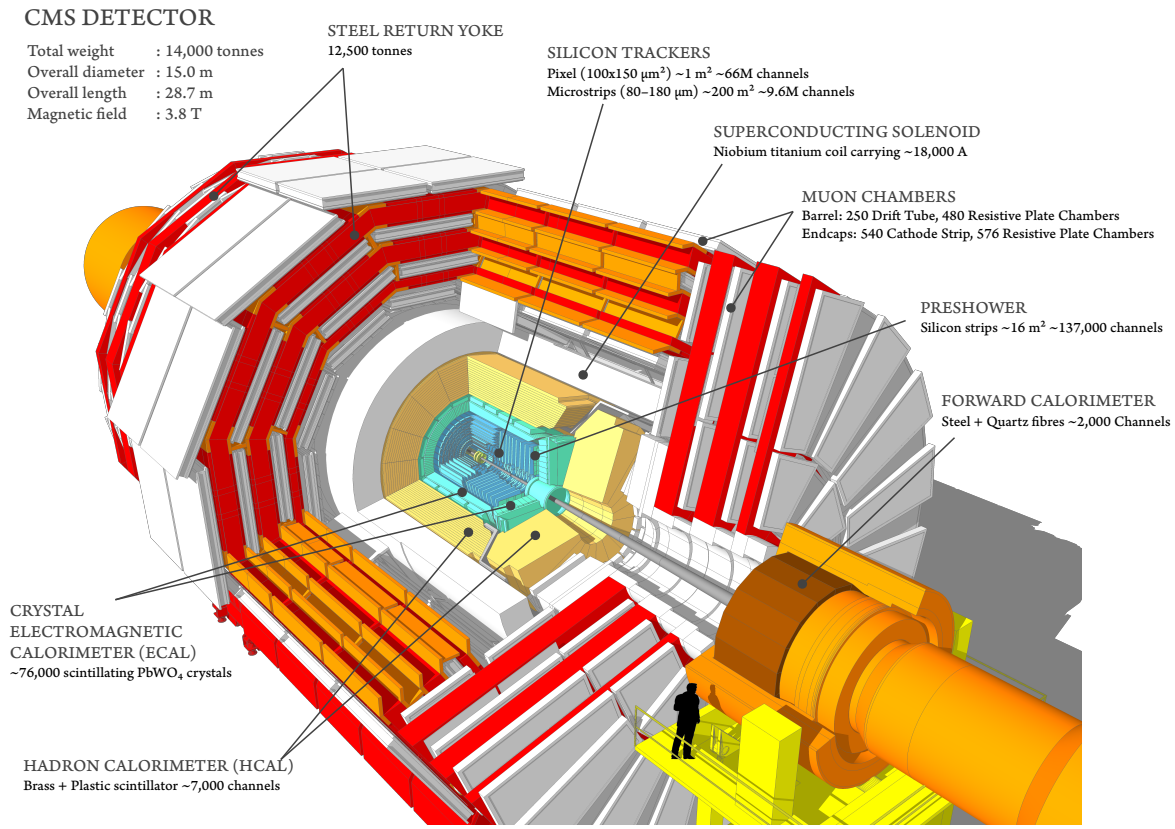


Figure 3.2.: Schematic overview of the CMS detector with its components. Taken from [21].

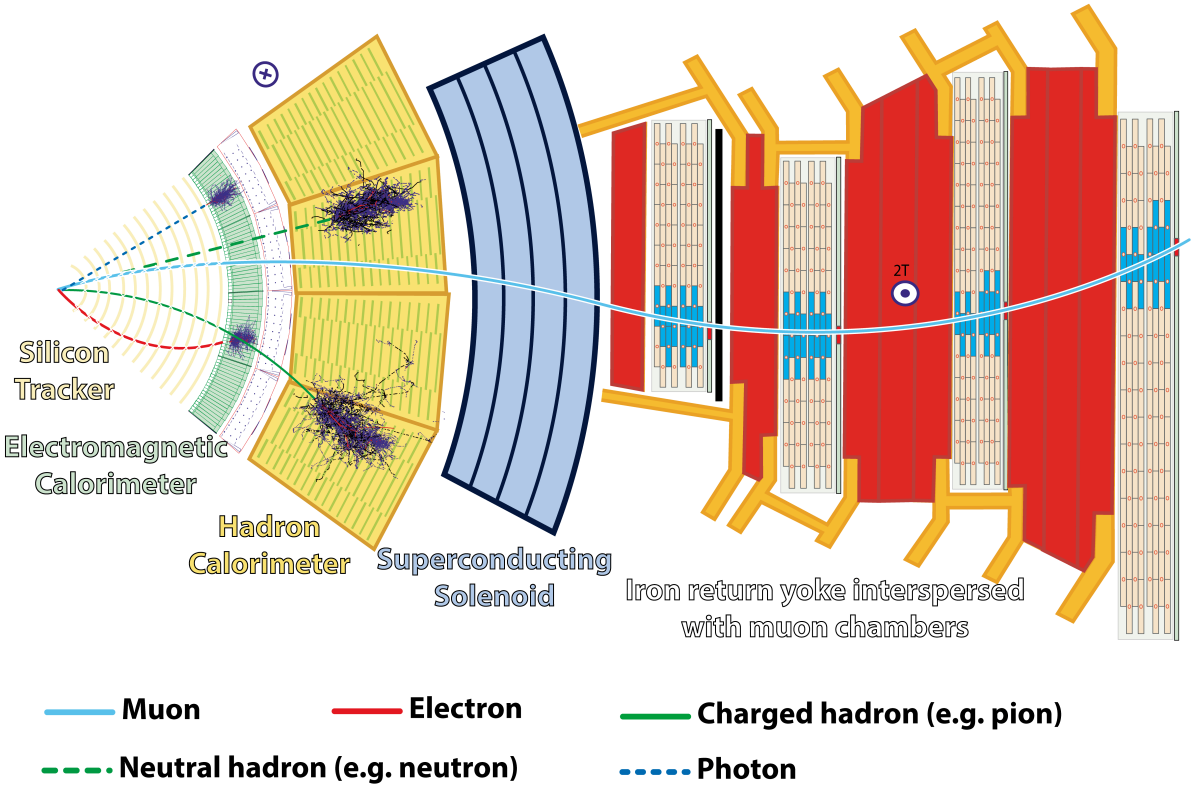


Figure 3.3.: Sketch of different particles flying through the detector (transverse slice). Taken from [22].

hadron calorimeter (HCAL), muon chambers and superconducting solenoid magnets. The magnets create a magnetic field of 3.8 T to bend the trajectories of charged particles. A schematic overview of the CMS detector is shown in Figure 3.2, where one can see, that it is cylindrical around the beam axis. It is also centred at the collision point. The detector can be divided into a barrel and a forward and backward part. The barrel part is the central region with $|\eta| \leq 3$. The forward and backward parts are the endcaps which measure particles with $3 \leq |\eta| \leq 5$. η is the pseudo-rapidity given by

$$\eta = -\ln[\tan(\frac{\theta}{2})] \tag{3.3}$$

where θ is the polar angle. In Cartesian coordinates, it is given by

$$\theta = \arctan(\frac{\sqrt{x^2 + y^2}}{z}) \tag{3.4}$$

In the relativistic limit, the pseudo-rapidity η converges to the rapidity y, defined as

$$y \equiv \frac{1}{2}\ln(\frac{E + p_z}{E - p_z}) \tag{3.5}$$

The rapidity y and the pseudo-rapidity η are used together with the azimuthal angle ϕ and the transverse momentum p_T in LHC-physics because they are Lorentz invariant. The following parts describe some of the detector components in more detail.

3.2.1. Silicon Tracker

The Silicon Tracker consists of pixel and strip detectors to reconstruct the trajectories of particles with $|\eta| \leq 2.5$. Closest to the beam axis are three layers of hybrid pixel detectors with a distance of 4.4 cm, 7.3 cm and 10.2 cm. These are important to reconstruct further primary vertices due to pile-up and secondary vertices from decaying b-quarks. The tracker needs to be as close as possible to the beam axis and interactions point to increase the resolution of the vertices. At a distance of 20 cm $\leq r \leq$ 116 cm, silicon microstrip detectors are built. In the forward and backward regions are two more layers of pixel and nine layers of microstrip detectors placed to detect particles with higher $|\eta|$.
In Figure 3.3, the silicon tracker and different particle trajectories through it are shown.

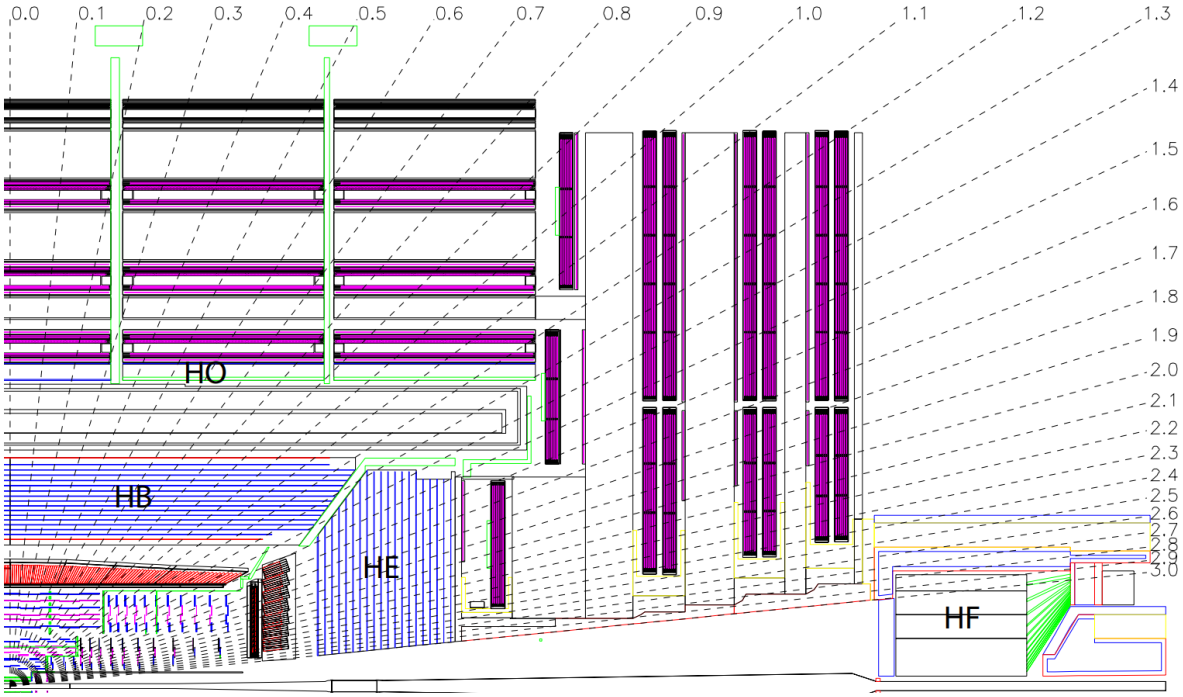


Figure 3.4.: Schematic view of the hadron calorimeter with its four parts: hadron barrel (HB), endcap (HE), outer (HO) and forward (HF) calorimeters. Taken from [16].

Due to the Lorentz force induced by the superconducting magnets, charged leptons and hadrons show a curved track.

3.2.2. Electromagnetic and Hadron Calorimeter

The Electromagnetic Calorimeter (ECAL) is mainly used to detect and measure electrons and photons, while the Hadronic Calorimeter (HCAL) is used to measure jet energies. The calorimeters are located around the tracker with the ECAL on the inside and the HCAL further outside. The calorimeters are located directly outside of the tracker, because the particles to be measured lose a part of their energy when flying through matter. Both calorimeters employ scintillation, where the energy of the particles detected in the calorimeter is proportional to amount of light gathered.
The ECAL consists of 61200 tungsten crystals in the barrel region with $|\eta| \leq 1.479$ and more than 7000 crystals in the endcaps with $|\eta| \leq 3$. The crystals are made out of lead tungstate ($PbWO_4$) crystals as scintillator material. It is a very dense material and therefore its radiation length, which is related to the energy loss of high energy particles electromagnetically interacting with it, is short. This is important, because it

reduces the space requirements of the ECAL. Additionally, the crystals are transparent, which is another requirement, because the crystals emit light when interacting with particles. A third requirement for the crystals is a quick light emission as new interactions happen every 25 ns. These tungsten crystals can emit 80 % of their light in this short timescale. The light is then collected via photodetectors, which are required to work with the low light yield from the crystals.
The HCAL can be divided into hadron barrel (HB), endcap (HE), outer (HO) and forward (HF) calorimeters, as seen in Figure 3.4. Different to the other three parts, which are placed between the ECAL and the magnet, the HO is built outside of the solenoid. The HF is closest to the beam-axis and covers a pseudo-rapidity range of $3 \leq |\eta| \leq 5$. The HE covers the range $1.3 \leq |\eta| \leq 3$, and the HB and HO can measure particles with $|\eta| \leq 1.4$ and $|\eta| \leq 1.3$, respectively. The HCAL consists of layers of scintillators and an absorber material in alternating order. Brass was chosen as the absorber material, as it is not magnetic and has a short hadronic interaction length, which is the equivalent to the radiation length of the ECAL material. However, the hadronic interaction length is much longer than the radiation length and therefore, the HCAL is larger than the ECAL. This is also the reason, why the HCAL is located outside of the ECAL, as the hadrons deposit only a small amount of their energy in the ECAL when flying through it. The scintillators are plastic scintillator tiles and are read out with photodetectors.
In Figure 3.3, the electron and photon being stopped in the ECAL and the hadrons being detected in the HCAL are shown.

3.2.3. Muon Chambers

Muons are detected in the muon chambers furthest on the outside of the detector because muon lose only a small amount of their energy when passing through matter. The CMS detector has four muon chambers in the barrel region interspersed with iron return yokes. They span a pseudo-rapidity region of $|\eta| \leq 2.4$. The muon chambers are aluminium drift tubes, consisting of gas filled chambers with a wire in the centre. An electrical field is then applied. When a muon (or any ionizing particle) passes through the chamber it ionized the gas. The ions and electrons are then separated by the electrical field and measured to determine the muon p_T. Higher muon rates are expected in the forward and backward region. Therefore, cathode strip chambers are used in the endcaps, as they have a higher response time. A third muon detector

are the resistive plate chambers, placed in both the endcaps and barrel region. The have a fast response time and are used to detect and exclude cosmic muons as well as determine the correct bunch crossing, as new interactions happen every 25 ns.
In Figure 3.3, the muon track is shown, it passes through the entire CMS detector without being stopped and is then detected in the muon chambers.

3.3. Reconstruction

The reconstruction of events from the information gained by the detector components is done by the Particle Flow (PF) Algorithm [23]. It identifies muons, electrons, photons and neutral and charged hadrons. It first reconstructs the information from each of the four detector subsections by itself, before linking the results together in the next step. In the final step, these linked results are used to identify the particles. The output of the PF algorithm can then be used in further steps, e.g. the Jet Clustering in Section 3.4.
The hits from the silicon tracker and muon chambers are combined and, iteratively, assigned to tracks. Each iteration step has looser requirements for the reconstructed tracks and previously assigned hits are not taken into consideration any more. The energy deposited in the ECAL and HCAL is clustered by starting with a local maximum and clustering it together with the energy deposited in the adjacent crystals. The tracks are then combined with the clusters on their trajectories and assigned to particles. Muons are reconstructed first, as only they leave a track in the muon chambers. These tracks are then removed from further consideration. Afterwards, electrons are identified via tracks in the silicon tracker and deposited energy in the ECAL and this information is then again removed from further reconstruction efforts. The remaining tracks are then assigned to charged hadrons and again combined with energy clusters on their trajectory and removed from further consideration. Finally, photons have only an ECAL cluster and neutral hadrons have clusters in both ECAL and HCAL.
This was only a short introduction into the reconstruction and the Particle Flow Algorithm and further information can be found in [23].

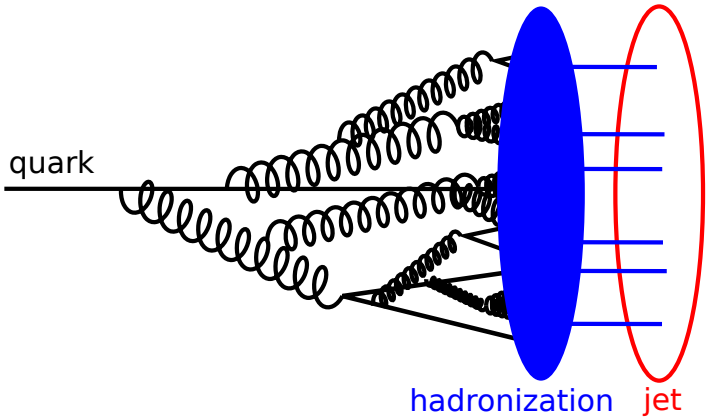


Figure 3.5.: Scheme for the creation of a jet from a quark.

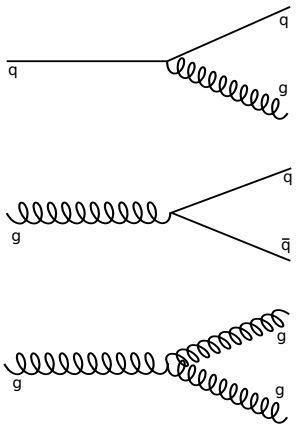


Figure 3.6.: Feynman diagrams involved in the creation of a jet.

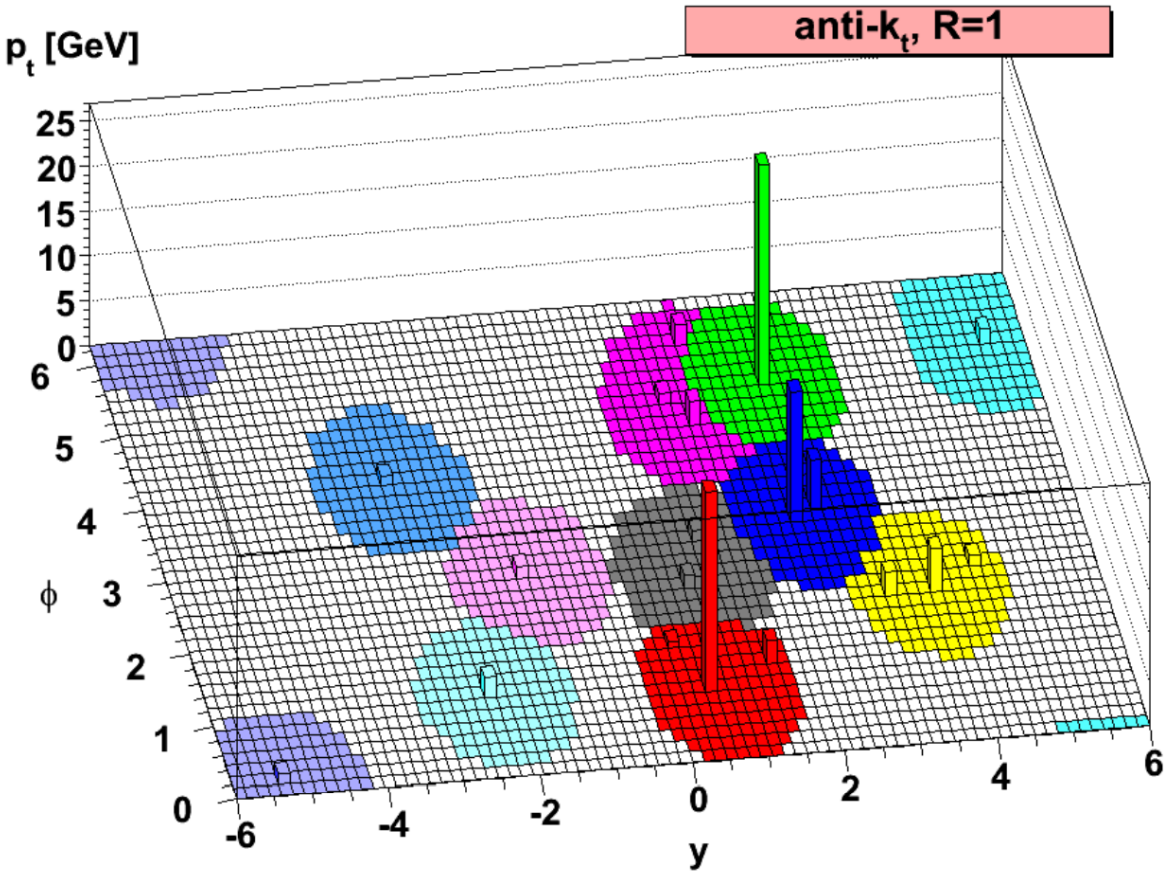


Figure 3.7.: Example reconstruction with the anti-k$_t$ jet clustering algorithm. Taken from [24].

3.4. Jet Clustering

Quarks and gluons created in the proton-proton collision have to hadronize as they can not exist freely. In Figure 3.5, a scheme for the creation of a jet is shown. The initial parton splits via the feynman diagrams displayed in Figure 3.6 into two partons, each having lower energy than the initial parton. Because the initial parton is relativistic, the secondary particles travel in the same general direction as the initial one. This happens repeatedly until the partons are below a certain energy threshold, where it becomes more favourable to form hadrons. These formed hadrons also contain unstable particles, which then decay.
These hadrons and their decay products are then detected. A jet clustering algorithm uses these particles to reconstruct the initial parton. However, this cannot be done precisely, because a jets are not well-defined. They are a localized cluster of particles, which depends on the choice of jet clustering algorithm.
A requirement for this algorithm is infrared and collinear (IRC) safety, which means that the result of the algorithm does not change when an infrared (low energy) particle is emitted or when a particle splits into two collinear particles. The standard algorithm used in the CMS collaboration and also in this thesis is the anti-k$_t$ algorithm [24]. As seen in Figure 3.7, this algorithm produces roughly circular jets with a radius R from Equation 3.7, approximately. The inputs from the PF algorithm into the anti-k$_t$ algorithm are called pseudo-jets. The anti-k$_t$ algorithm clusters jets based on the distance between a pseudo-jet and the beam d_{iB} and the distance between two pseudo-jets d_{ij}. The distances are given by

$$d_{iB} = p_{Ti}^{-2} \tag{3.6}$$

and

$$d_{ij} = min(p_{Ti}^{-2}, p_{Tj}^{-2}) * \frac{\Delta R_{ij}^2}{R^2} \tag{3.7}$$

with

$$\Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2 \tag{3.8}$$

In these equations, p_{Ti}^2 is the transverse momentum of pseudo-jet i, ΔR_{ij}^2 the distance between pseudo-jets i and j, y_i and ϕ_i are the rapidity and the azimuthal angle of

particle i and R is a free parameter. In this thesis, $R = 0.4$ is chosen and therefore, the resulting jets are called AK4-jets.
The anti-k_t algorithm first calculates all d_{iB} and d_{ij} and identifies the smallest. If it is d_{iB}, the pseudo-jet i is called a jet and taken out of further calculations. If it is d_{ij}, the pseudo-jets i and j are combined. These steps are done iteratively until no more pseudo-jets are left. Due to these settings, the clustering starts around a local p_T-maximum by combining it with its neighbouring lower p_T-particles, as seen in Figure 3.7.
This Section is based on [24].

Chapter 4.

Input Measurements

In this chapter, the jet substructure variables are presented in Section 4.1. Section 4.2 describes the differences of dijet and Z+jet events and Section 4.3 gives an overview of the sources of uncertainties based on [4]. Finally, Section 4.4 gives an introduction to the underlying event measurement.

4.1. Jet Substructure Variables

The jet substructure variables, also called generalized angularities, studied in this thesis are defined as

$$\lambda_\beta^\kappa = \sum_{i \in jet} z_i^\kappa \theta_i^\beta \tag{4.1}$$

where z_i is the momentum fraction and θ_i is the normalized angle to the jet axis of particle i. They are defined as

$$z_i \equiv \frac{p_{Ti}}{\sum_{j \in jet} p_{Tj}}, \quad \theta_i \equiv \frac{R_{i\hat{n}}}{R} \tag{4.2}$$

(0,0)	(2,0)	(1,0.5)	(1,1)	(1,2)
multiplicity	$(p_T^D)^2$	LHA	width	thrust

Table 4.1.: Overview of the five jet substructure observables where (0,0) refers to the (κ,β) variables.

where p_{Ti} is the transverse momentum of particle i and $R_{i\hat{n}}$ is the rapidity-azimuth distance of particle i to the jet axis. In this thesis, five jet substructure observables are used, presented in Table 4.1. The multiplicity and $(p_T^D)^2$ are not infrared and collinear (IRC) safe. IRC safe means, that the variable is not affected when an infrared (low energy) particle is emitted or when a particle splits into two collinear particles. This is important, because the splitting is proportional to E^{-1} and θ^{-1} and, therefore, a lot of infrared and collinear splitting happens. However, these non-IRC safe variables, have been used in existing quark- and gluon-jet discriminators, and have often shown the largest difference. The LHA, width and thrust are IRC safe. They differ in their weighting of the particles. The jet substructure observable with $\beta = 0.5$ weights particles higher, if they are closer to the centre of the jet, while a $\beta = 2$ means, that particles away from the centre have a higher weight.
Further information on the jet substructure variables can be found in [2] and [3].

4.2. Dijet and Z+Jet Events

The Z+jet selection requires at least one reconstructed muon with p_T >26 GeV, as the Z boson is reconstructed from Z$\rightarrow \mu^+\mu^-$ final states. Two oppositely charged muons with $|\eta| < 2.4$ and $\left|M_{\mu\mu} - M_Z\right|$ <20 GeV, with the di-muon invariant mass $M_{\mu\mu}$ and the Z boson mass M_Z, are required. At least one reconstructed jet with p_T >30 GeV and $|y| < 1.7$ is required.
For the dijet selection, at least two reconstructed jets with p_T >30 GeV and $|y| < 1.7$ are required. The transverse momentum asymmetry is required to be $(p_T^{j_1} - p_T^{j_2})/(p_T^{j_1} + p_T^{j_2}) < 0.3$, where $p_T^{j_1}$ and $p_T^{j_2}$ are the p_T of the two jets with the largest p_T. These two jets are then selected. This selection is applied in the analyses created by Robin Aggleton [4], which are displayed in Appendices B.1 and B.2.
Figure 4.1 shows the fraction of gluon-jets for dijet and Z+jet event jets versus the p_T of the jet. Although it is impossible to measure and define a jet as definitely originating from a quark or gluon, it is possible to select processes that are quark- or gluon-jet enriched. Z+jet event jets are more likely to originate from a quark over the entire p_T spectrum, while dijet event jets are more likely to originate from a gluon in the lower p_T-bins up to around 300 GeV. Figures 4.2 and 4.3 display the leading order Feynman diagrams for a dijet or Z+Jet event, respectively. It can be seen, that both dijet and Z+jet events can have either quarks or gluons in the final state.
The dijet event jets are split into a more central (smaller $|\eta|$) and forward (larger $|\eta|$)

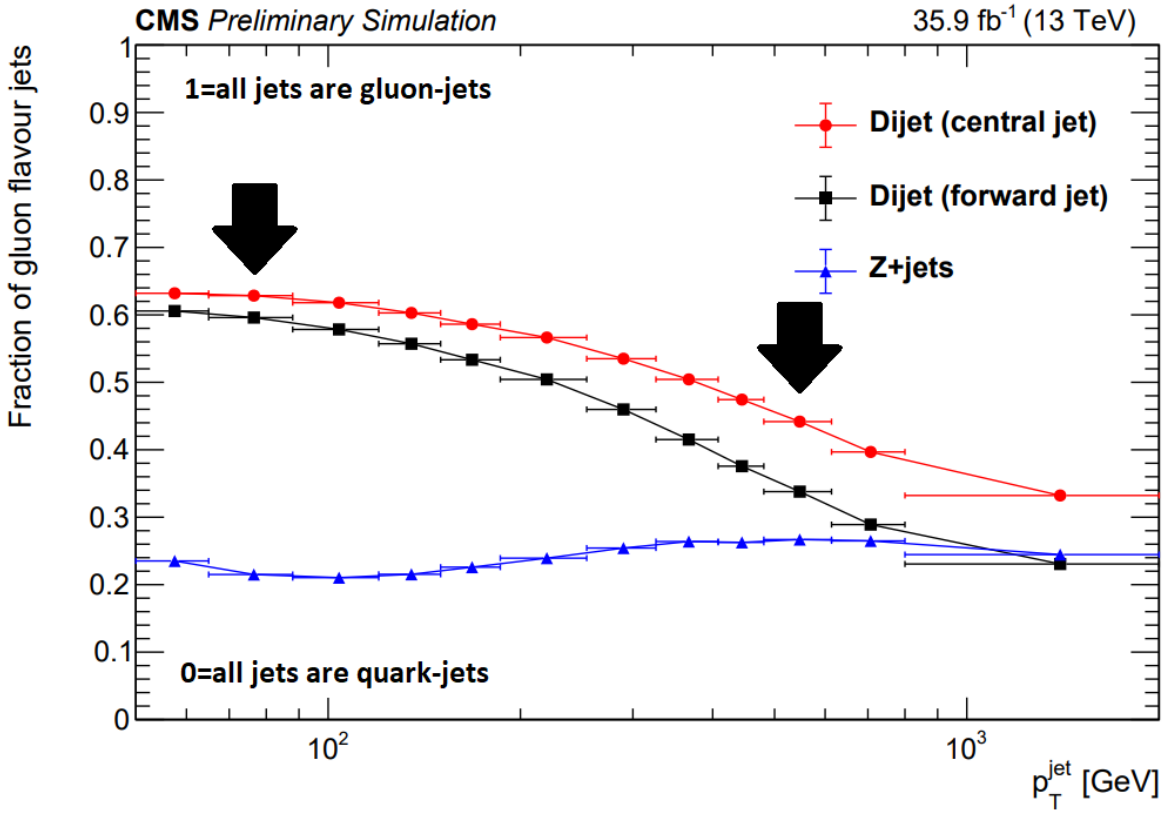


Figure 4.1.: Fraction of gluon-jets in Z+jet and dijet event samples. Dijet event jets are split into central and forward region. Black arrows mark the p_T-bins for the plots presented in Chapter 6. Taken from [4].

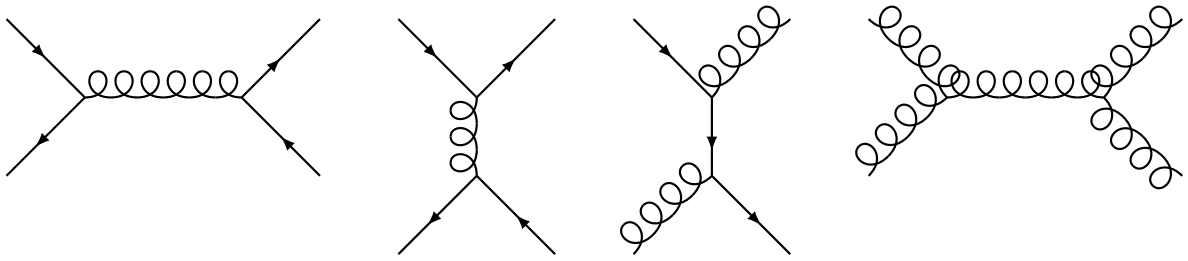


Figure 4.2.: Leading Order Feynman diagrams of a dijet event. The straight and spiral lines correspond to quarks and gluons, respectively.

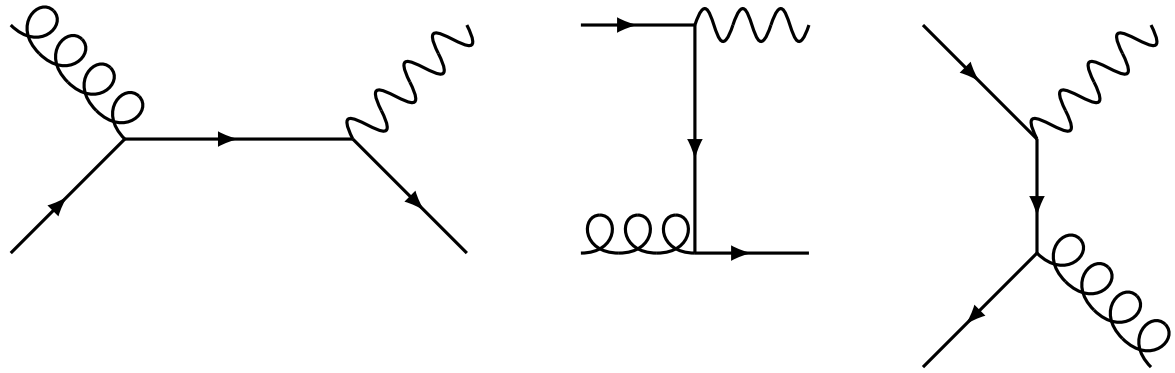


Figure 4.3.: Leading Order Feynman diagrams of a Z+jet event. The straight and spiral lines correspond to quarks and gluons, respectively, and the wavy lines to the Z boson.

jet. The central jet is more likely to originate from a gluon and has a bigger difference to Z+jet event jets than the more forward jet. Therefore, only the more central jet of a dijet event is studied in this thesis.
Quark-jets have been more used in tuning, therefore, studying gluon-jets is more interesting. For this reason, the 65 GeV to 88 GeV pT-bin is studied for both Z+jet and dijet (central jet) events and the 481 GeV to 614 GeV p_T-bin is only studied for dijet events.

4.3. Uncertainties

In the data taken in [4], various sources of systematic uncertainties are taken into account. They can be split into experimental sources, uncertainties from the physics model and unfolding uncertainties. They are treated in various ways described in [4]. For example, uncertainties in the measurement of the PF algorithm particles are taken into account for the calculation of the jet substructure observables, including a $\pm 1\,\%$ uncertainty for the charged hadron momentum and the photon energy scale and a $\pm 3\,\%$ uncertainty for the neutral hadron energy scale. The effect of pile-up on the uncertainties of the jet substructure observables is estimated by reweighting events within the uncertainty of the pileup distribution. The largest effect originates from using an alternate shower and hadronization model from Herwig++.
The total uncertainty also includes statistical uncertainties, which are smaller for dijet events, but have a larger effect in Z+jet events.

4.4. Underlying Event Measurements

In this thesis, two underlying event analyses were used: **CMS_2015_I1384119** [25] and **CMS_Internal_FSQ_15_007** [26].
The first analysis measures the pseudo-rapidity distribution of charged hadrons in the central ($|\eta| \leq 2$) detector region. The dependence on the collision energy of this distribution reflects the contribution of soft- and hard-scattering on the hadron production. Soft interactions, only described by phenomenological models, lead to a large fraction of these particles. The fraction of particles from hard interactions increase with increasing collision energy. The measurement of this distribution is used to tune the modelling of MC generators.
The second analysis measures the average charged-particle multiplicity density (number of tracks divided by the area of the region) and the average energy density (sum of track p_T divided by the area of the region). They are measured in four regions: transMax and transMin, transDif, which is the difference of transMax and transMin densities, and transAve, which is the average density of both transverse regions. The transMax and transMin regions were defined in Section 2.3. They are binned depending on the p_T of the leading track. The measurement of these densities is used to tune the modelling of multi-parton interactions with MC generators.

Chapter 5.

Analysis

In this chapter, the varied parameters for the simulation with Pythia8 and Herwig7 are presented in Section 5.1. Section 5.2 is based on [27] and gives a short introduction to Rivet, the toolkit used for the simulation of the jet substructure with Pythia8 and Herwig7 and the creation of the plots. The analyses used for this thesis were created by Robin Aggleton [4] and are displayed in Appendices B.1 and B.2.

5.1. Relevant Parameters

In this thesis, the effects of varying α_S and Colour Reconnection (CR) modes are examined. As a baseline tune, the CP5 tune [14] is used for Pythia8. For Herwig7, the CH2 tune [13] is used as baseline. Both baseline tunes were tuned to the UE measurements. Therefore, the underlying event plots in Section 6.5 should show the best agreement with the default $\alpha_S = 0.118$. The baseline tunes can be found in Appendix C. Further information about the settings for Pythia8 and Herwig7 can be found in [28] and [29], respectively.

5.1.1. α_S

For Pythia8, the variable **TimeShower:alphaSvalue** is varied. The default value is $\alpha_S = 0.118$ [14]. **TimeShower** corresponds to the final state radiation of the partons, so varying this α_S value, only the parton shower of the outgoing partons is affected. For Herwig7, the variable **AlphaQCD:AlphaMZ** is varied. Herwig7 has the same default value for the CH2 tune of $\alpha_S = 0.118$ [13]. Varying this value changes α_S for all

strong interactions. Therefore, not only the final state radiation is affected, but also, for example, the initial state radiation.
For both generators, $\alpha_S = 0.113$, 0.118 and 0.123 are studied. Therefore, this is not a fine-tuned scan over α_S as this thesis is only designed to show how varying α_S affects the jet substructure simulation.

5.1.2. Colour Reconnection modes

For the variation of the CR modes for Pythia8, the variable **ColourReconnection:mode** needs to be changed to 0, 1 or 2 for the CR modes studied in this thesis. Additionally, as these CR modes have different settings, these also needs to be adjusted. In this thesis, the default CP5 tunes have been used for all three CR modes. The settings can be found in the Pythia8 configuration files in Appendix C.
CR mode 0 is the MPI-based model and it is the original CR mode. In this model, the total string length is reduced by inserting the gluons of a lower-$p_\perp$ interaction into a colour dipoles of a higher-$p_\perp$ interaction. CR mode 1 is the QCD-based model, where the total string length is reduced by identifying alternative coherent parton–parton states beyond leading colour and the possibility of forming junctions. CR mode 2 is the gluon-move model, that moves gluons on a colour line connecting a parton pair, if doing so results in a reduction of the total string length. In a second step, two colour connected parton pairs can reconnect to exchange the colour connected partner. Further information on the three CR modes can be found in [11], [30] and [31].

5.2. Rivet

Rivet (Robust Independent Validation of Experiment and Theory) is a toolkit used for the validation of Monte Carlo event generators. It contains, among others, an infrastructure to create analyses with C++ as well as a large collection of validated analyses from previous studies. Different MC generators can be plugged in via Python scripts. It has been used for developing, validating and tuning of MC generators for SM processes as well as the development of new analysis techniques, e.g. jet substructure analysis. For further information, please refer to [27], [32]and [33].
In this thesis, the **Rivet release 2.7.2** was used. In Appendix C, the configuration files for the Rivet toolkit are presented.

Chapter 6.

Results and Discussion

In the following sections, the results are presented. In Section 6.1, α_S comparisons are shown for both generators to see how it affects the jet substructure and which α_S-value performs better. The effects of α_S comparisons for different Colour Reconnection (CR) models in Pythia8 are shown in Section 6.2. In Section 6.3, the performance of Pythia8 and Herwig7 for jet substructure is compared. A comparison of the description of quark- and gluon-jets is presented in Section 6.4. The effects of the α_S-variations on the underlying event tune is shown in Section 6.5.
The following abbreviations are used in the legends of the plots in this chapter: Pythia8 with CR mode 0 of the CP5 tune (P8 CR0), Herwig7 CH2 tune (H7 CH2), dijet event jets (DIJET) and Z+jet event jets (ZPJ).

6.1. α_S Comparison

Figures 6.1 to 6.4 show the results for Pythia8 with CR mode 0. In Figure 6.1, the jet substructure variable LHA for Pythia8 dijet events in the high p_T-bin with α_S variations is shown. On the x- and y-axis, the λ_β^κ-variable and the number of events in this p_T- and $\lambda_{0.5}^1$-bin are displayed, respectively. The red line with the yellow error region is the data. The three α_S-values 0.113, 0.118 and 0.123 are the blue, green and yellow line, respectively, with $\alpha_S = 0.118$ being the default value. In the bottom plot, the ratio *Pythia8/data* is displayed for all three α_S-values.
The simulation from Pythia8 describes the data well and depicts both the peak and the tail of the data. Clearly visible in both plots is, that $\alpha_S = 0.123$ performs better than the other two α_S-values over the entire range of the plots. The default value $\alpha_S = 0.118$

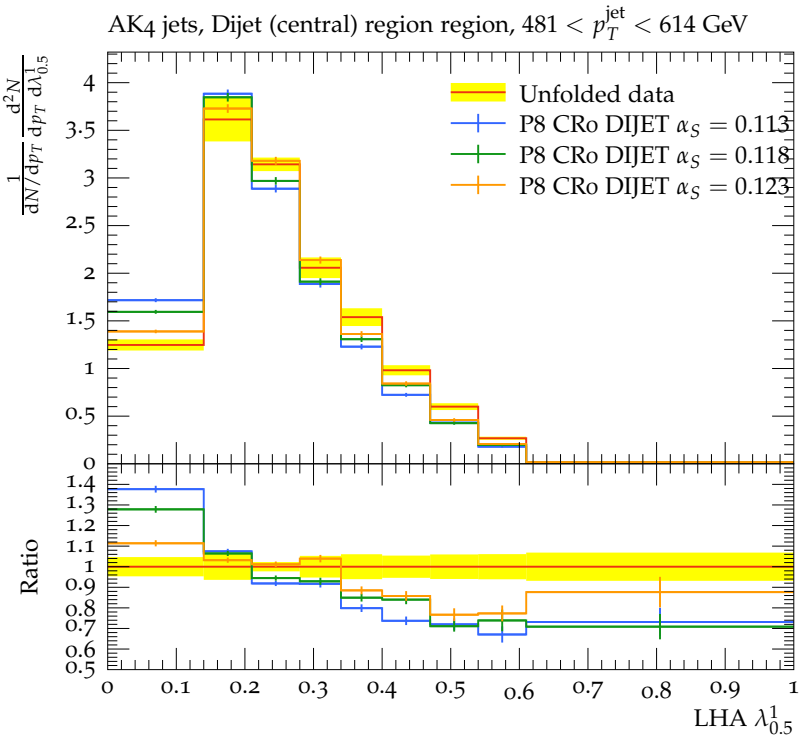
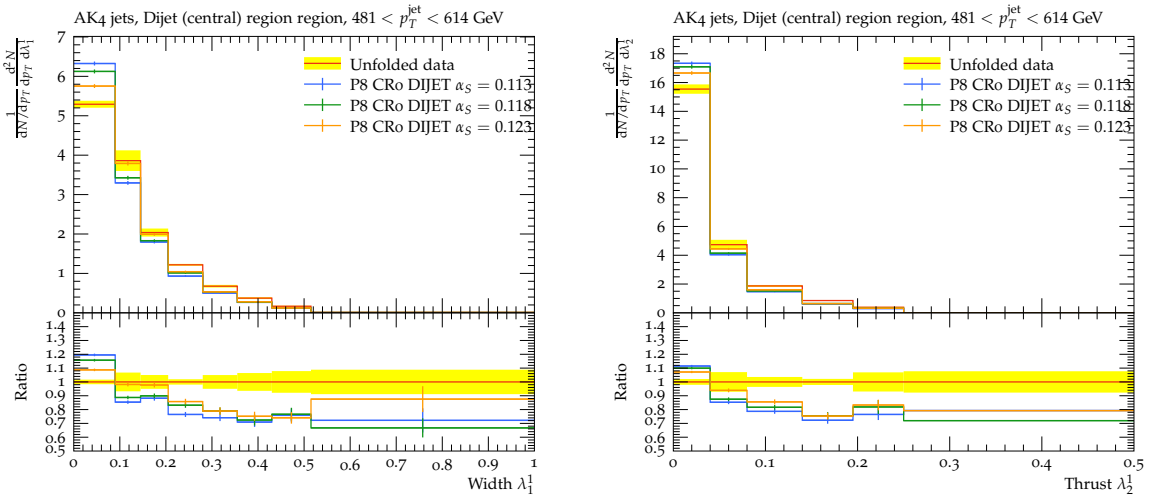


Figure 6.1.: LHA ($\lambda^1_{0.5}$) for Pythia8 dijet events in the high p_T-bin with α_S variation

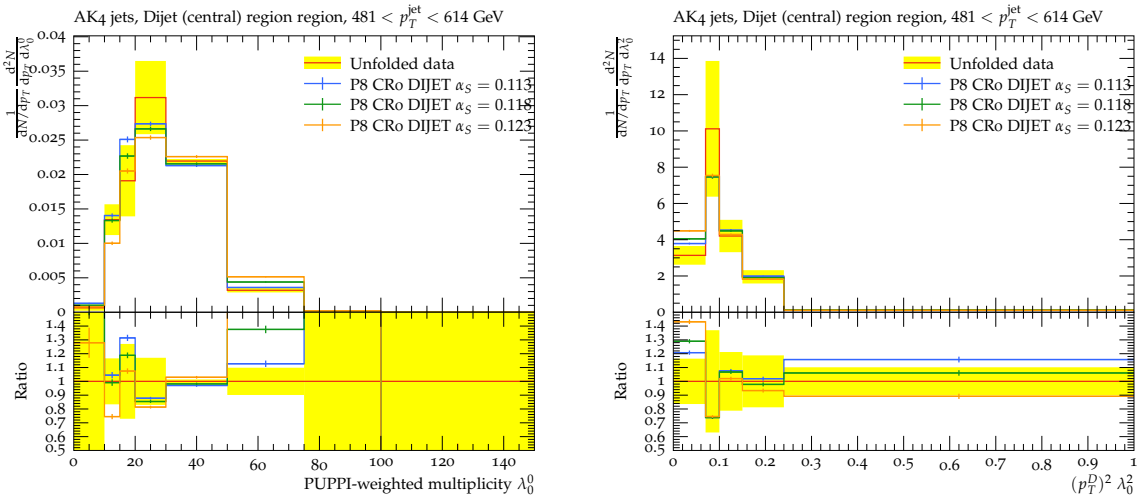


(a) Width (λ^1_1) **(b)** Thrust (λ^1_2)

Figure 6.2.: Pythia8 dijet events in the high p_T-bin with α_S variation

performs better than $\alpha_S = 0.113$, this indicates, that higher α_S-values describe this substructure variable better.

In Figure 6.2a and 6.2b, the width and thrust substructure variables in the high p_T-bin are displayed, respectively. Both λ^κ_β-variables are well described by Pythia8



(a) Multiplicity (λ_0^0) **(b)** $(p_T^D)^2$ (λ_0^2)

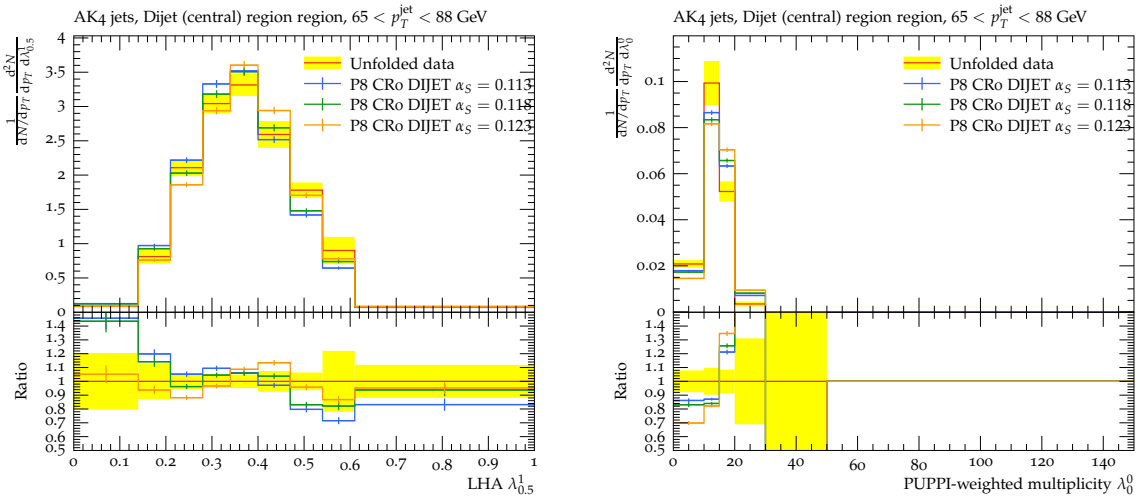
Figure 6.3.: Pythia8 dijet events in the high p_T-bin with α_S variation

and $\alpha_S = 0.123$ performs the best and $\alpha_S = 0.113$ the worst out of the three α_S-variables. The results from these two substructure variables matches with the outcome of Figure 6.1 with the LHA variable. In fact, these three substructure variables always perform similar and, therefore, only the the LHA variable will be used in this thesis. The plots with width and thrust can be found in Appendix A.1.

In Figures 6.3a and 6.3b, the multiplicity and $(p_T^D)^2$ substructure variables in the high p_T-bin are displayed, respectively. These two variables are more difficult to model for Pythia8 and have the largest uncertainties among the studied substructure observables. Pythia8 underestimates the peak in both λ_β^κ-variables. In these two substructure variables, the effect of α_S on the jet substructure can be seen. A higher α_S leads to more interactions and, therefore, more particles with less energy per particle. This corresponds to fewer events in the first few bins for the multiplicity and more events in the later bins with a higher α_S. For the $(p_T^D)^2$, it is inverse to the effect on the multiplicity. In Figures 6.3a and 6.3b, this effect is clearly visible.

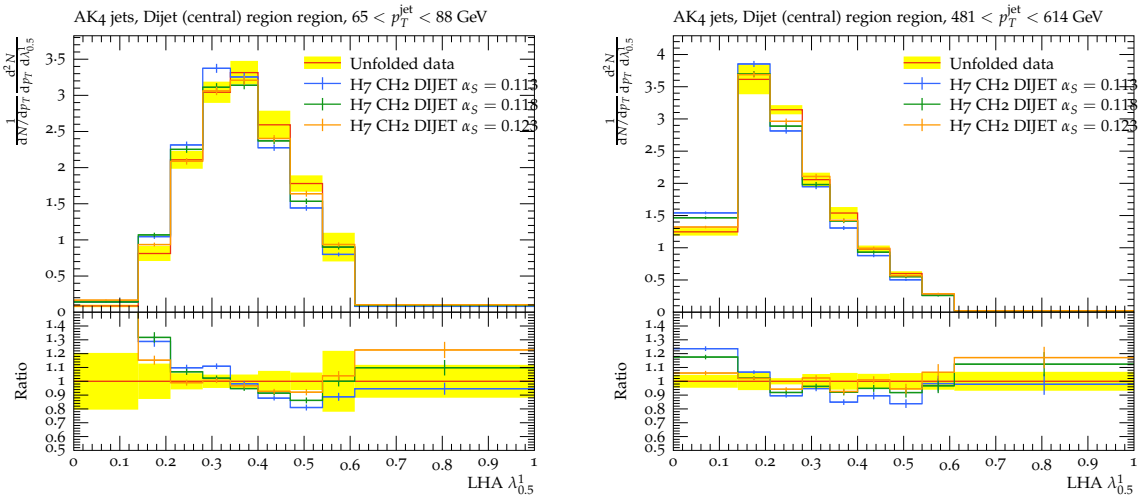
In these plots, there is no best α_S value over the entire range of the plots visible. When these two substructure variables perform similar, only one of them will be shown. All plots can be found in Appendix A.1.

In Figure 6.4a, the LHA variable in the low p_T-bin is shown. Pythia8 describes the data well in both the peak and the tails. In this p_T-bin, there is no clear best performing α_S value. In the tails, $\alpha_S = 0.123$ performs better, but in the peak region all three α_S values perform similar. In Figure 6.4b, the multiplicity is displayed. In this bin, the simulation from Pythia8 has a shifted peak compared to the data. None of the three α_S



(a) LHA ($\lambda^1_{0.5}$) **(b)** Multiplicity (λ^0_0)

Figure 6.4.: Pythia8 dijet events in the low p_T-bin with α_S variation

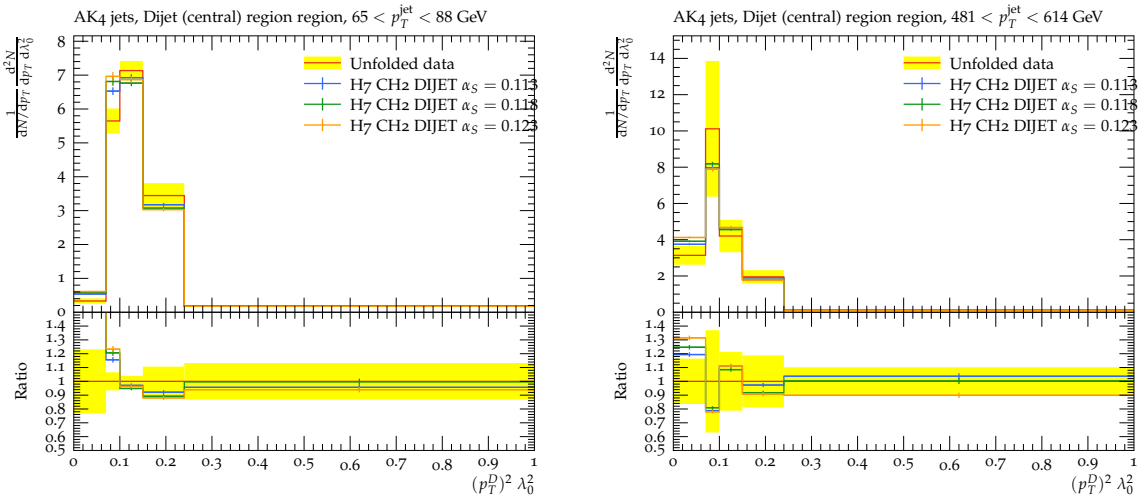


(a) low p_T-bin **(b)** high p_T-bin

Figure 6.5.: LHA ($\lambda^1_{0.5}$) for Herwig7 dijet events with α_S variation

values performs well, but $\alpha_S = 0.123$ performs a bit worse than the other two in this plot.

Figures 6.5 and 6.6 show the results for Herwig7. They display both p_T-bins for LHA and $(p_T^D)^2$, respectively. For the LHA variable, Herwig7 describes the data well in both p_T-bins over the entire range of the plot. $\alpha_S = 0.123$ performs the best in both p_T-bins besides the last two λ^κ_β-bins. The $(p_T^D)^2$ variable is, similar to Pythia8, difficult to model and has the largest uncertainties. Again, there is no clear best α_S value for this substructure variable.



(a) low p_T-bin **(b)** high p_T-bin

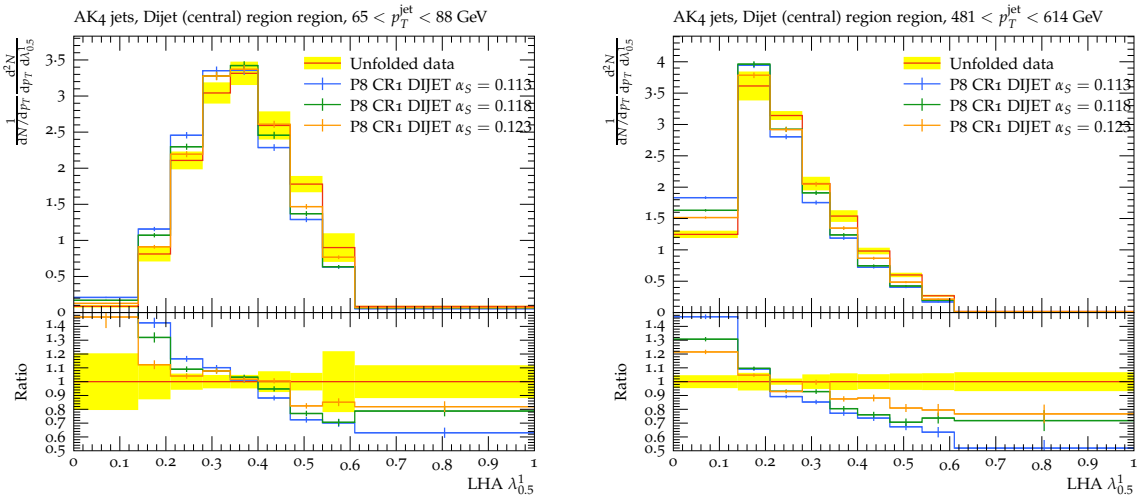
Figure 6.6.: $(p_T^D)^2$ (λ_0^2) for Herwig7 dijet events with α_S variation

Overall, Herwig7 and Pythia8 model LHA, width and thrust well, but multiplicity and $(p_T^D)^2$ are difficult to describe. For both generators, the simulation of jet substructure performs better with $\alpha_S = 0.123$ compared to the default value. However, as this was not a fine-tuned value, it can only be said, that an $\alpha_S > 0.118$ performs the best. Additionally, it is not possible to simulate the data well in all p_T-bins and substructure variables when varying only α_S.

6.2. Colour Reconnection Comparison

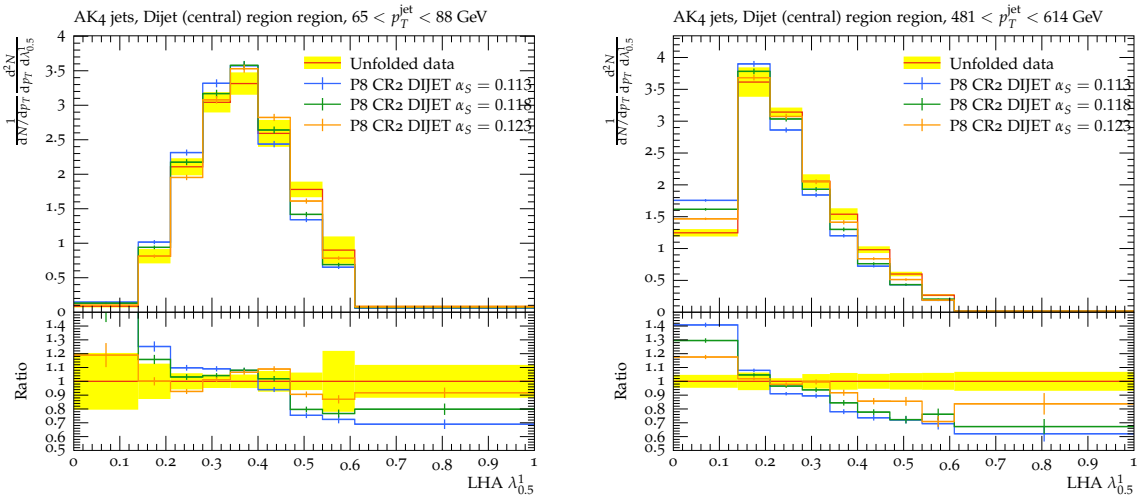
Figures 6.7 to 6.10 show the results for Pythia8 with CR modes 1 and 2. Again, the LHA variable in Figures 6.7 and 6.8 is well described by Pythia8 and the multiplicity in Figures 6.9 and 6.10 is more difficult to model, especially in the low p_T-bin. For the high p_T-bins in Figures 6.7b and 6.8b, $\alpha_S = 0.123$ performs the best for the LHA substructure variable out of the three α_S values. However, different to CR mode 0, in the low p_T-bins for CR modes 1 and 2 in Figures 6.7a and 6.8a, $\alpha_S = 0.123$ also performs the best.

In further chapters, only CR mode 0 plots are shown, as neither CR mode 1 nor 2 show a superior performance and give new information compared to CR mode 0. However, both CR modes are interesting for further studies involving the fine-tuning



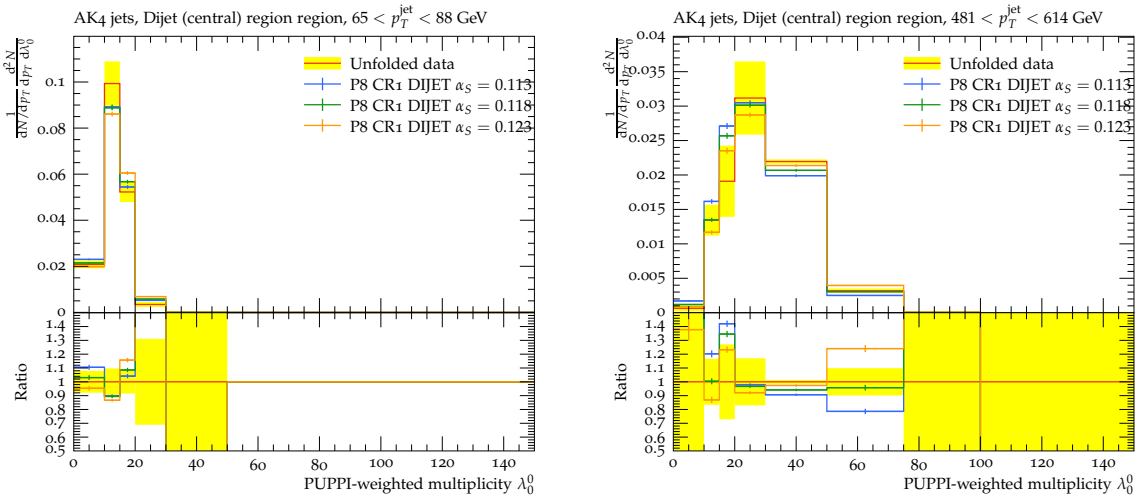
(a) low p_T-bin **(b)** high p_T-bin

Figure 6.7.: LHA ($\lambda^1_{0.5}$) for Pythia8 CR mode 1 dijet events with α_S variation



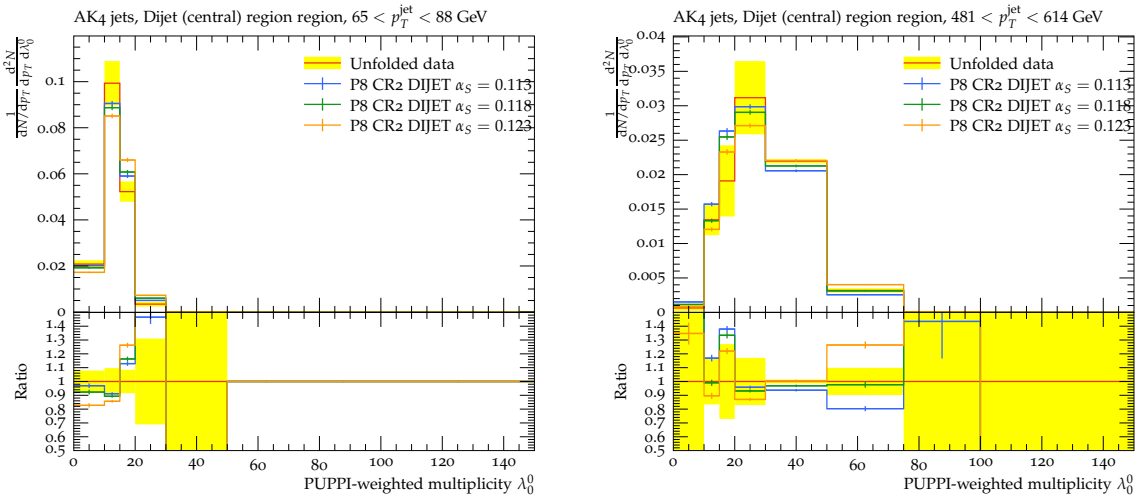
(a) low p_T-bin **(b)** high p_T-bin

Figure 6.8.: LHA ($\lambda^1_{0.5}$) for Pythia8 CR mode 2 dijet events with α_S variation



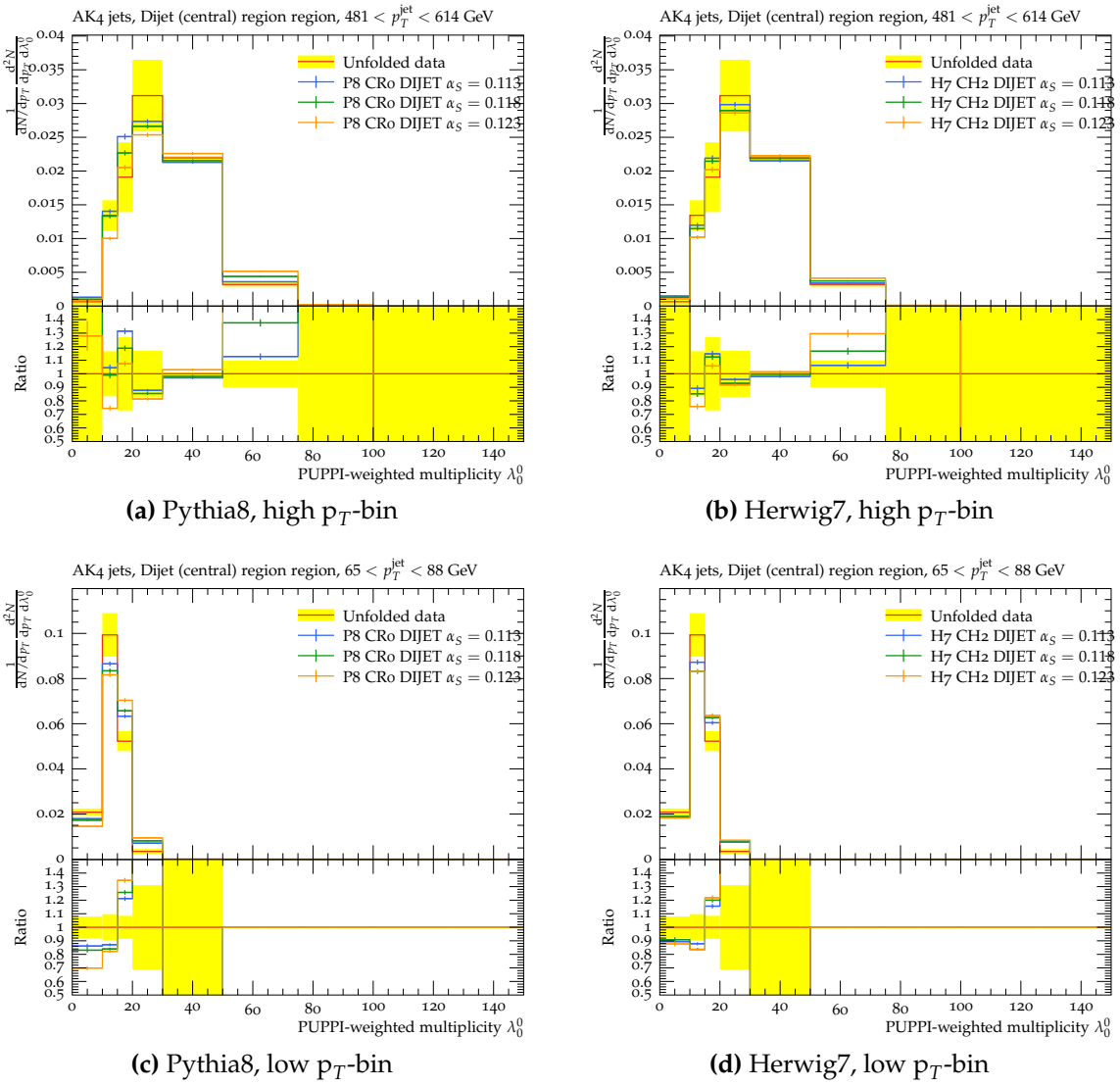
(a) low p_T-bin **(b)** high p_T-bin

Figure 6.9.: Multiplicity (λ_0^0) for Pythia8 CR mode 1 dijet events with α_S variation



(a) low p_T-bin **(b)** high p_T-bin

Figure 6.10.: Multiplicity (λ_0^0) for Pythia8 CR mode 2 dijet events with α_S variation



(a) Pythia8, high p_T-bin

(b) Herwig7, high p_T-bin

(c) Pythia8, low p_T-bin

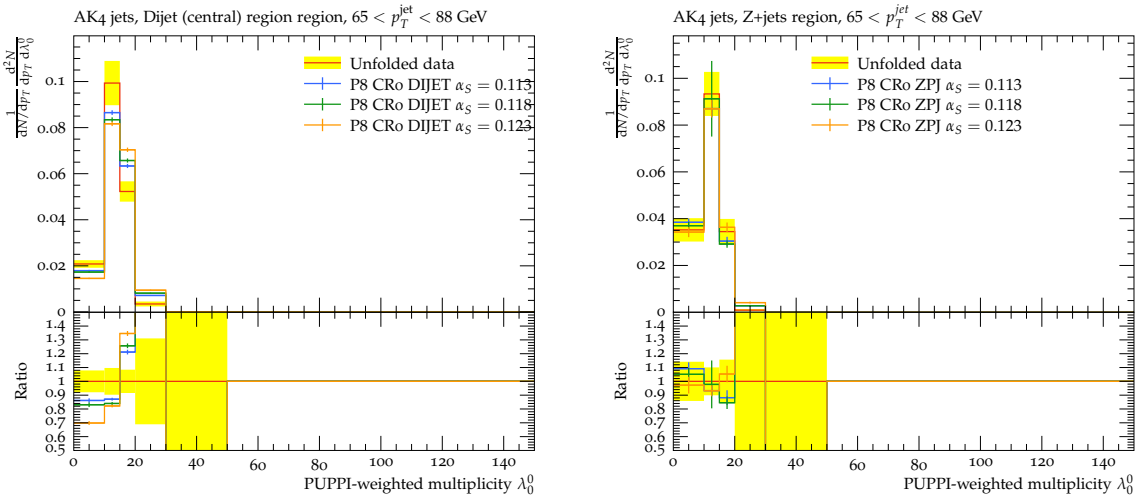
(d) Herwig7, low p_T-bin

Figure 6.11.: Multiplicity (λ_0^0) for dijet events with α_S variation

of α_S combined with other variables to these jet substructure variables as they might show a better performance.

6.3. Generator Comparison

The four plots in Figure 6.11 display the multiplicity for both Pythia8 and Herwig7 in both low and high p_T-bins. In the high p_T-bin, Herwig7 shows a better simulation of the data. Especially the peak is much better approximated by Herwig7. In the $20 \leq \lambda_0^0 \leq 30$ bin, Pythia8 shows a difference of 20 % in the ratio plot while Herwig7



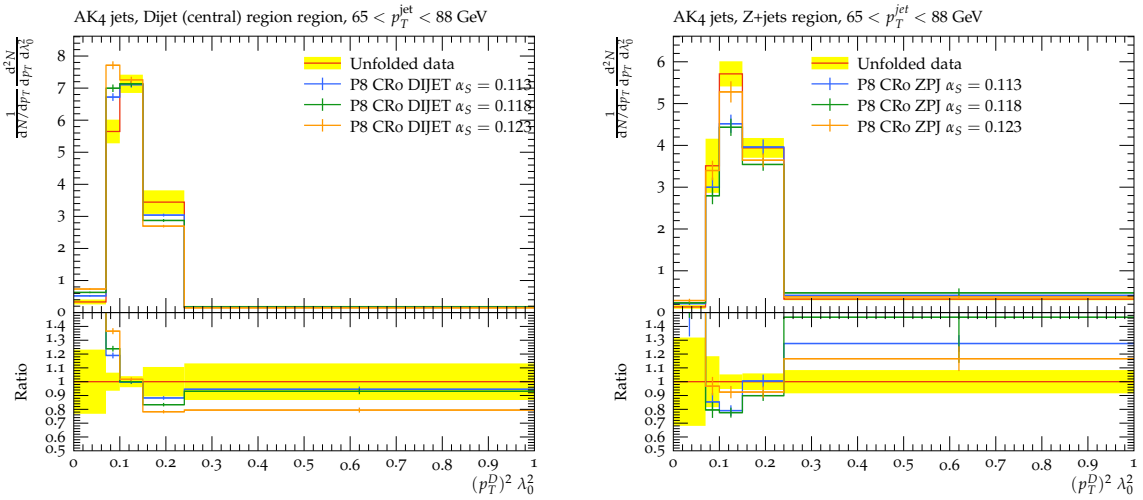
(a) Dijet events **(b)** Z+jet events

Figure 6.12.: Multiplicity (λ_0^0) for Pythia8 in the low p_T-bin with α_S variation

has less than a 10 % difference between simulation and data. However, in the low p_T-bins in Figures 6.11c and 6.11d, both generators perform similar and have difficulties to model the data.
Overall, Herwig7 performs at least similarly well and in some jet substructure and p_T-bins better compared to Pythia8. This might imply, that Herwig7 is better suited to model jet substructure. However, it is important to note, that the α_S variable varied in each of the generators have differences in their effects: for Pythia8, it only affects the final state radiation, while it affects every step of the generation process for Herwig7.

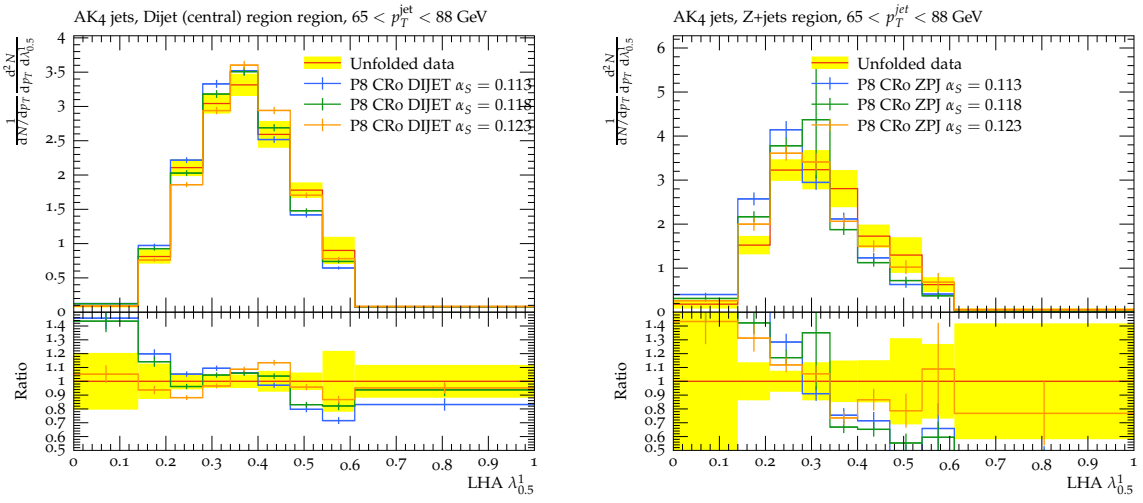
6.4. Quark- and Gluon-Jet Comparison

Figures 6.12 to 6.14 show comparisons of dijet and Z+jet event jets in the low p_T-bin for the jet substructure variables multiplicity, $(p_T^D)^2$ and LHA, respectively. The multiplicity for Z+jet event jets in Figure 6.12b is well described by Pythia8. All three α_S values perform similarly well and, therefore, no best α_S value can be determined. Compared to the dijet event jets in Figure 6.12a, this quark jets are clearly better described by Pythia8 in this jet substructure variable. The $(p_T^D)^2$ variable in Figure 6.13b is more difficult to model, as the peak is underestimated by Pythia8. However, an improvement for $\alpha_S = 0.123$ is clearly visible. In comparison with Figure 6.13a, the peak is not shifted to the left for Z+jet event jets and for $\alpha_S = 0.123$, the quark jets a slightly better



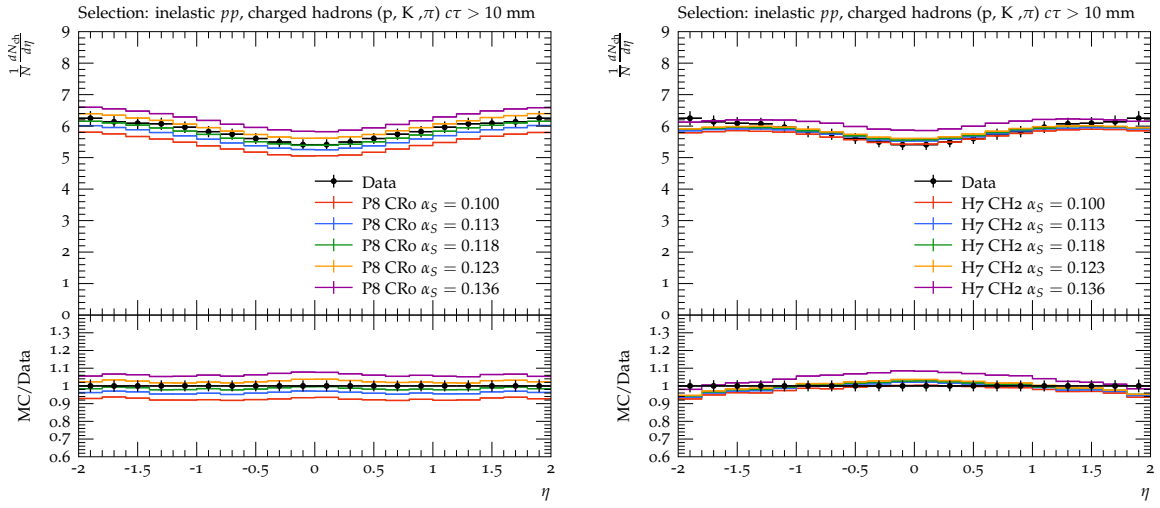
(a) Dijet events **(b)** Z+jet events

Figure 6.13.: $(p_T^D)^2$ (λ_0^2) for Pythia8 in the low p_T-bin with α_S variation



(a) Dijet events **(b)** Z+jet events

Figure 6.14.: LHA $(\lambda_{0.5}^1)$ for Pythia8 in the low p_T-bin with α_S variation



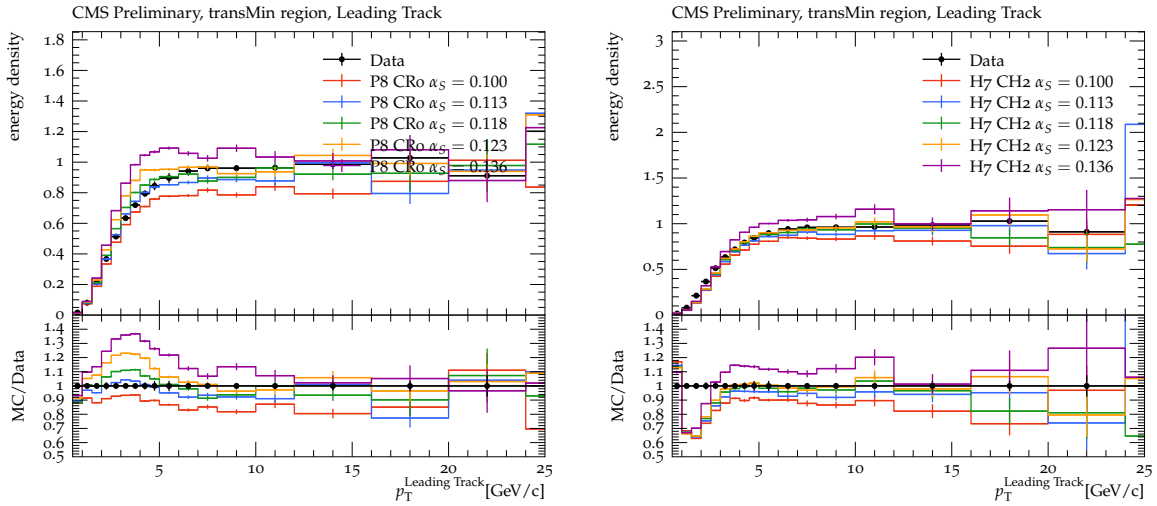
(a) Pythia8 **(b)** Herwig7

Figure 6.15.: Number of charged hadrons per η-bin with α_S variation

described. In Figure 6.14b, $\alpha_S = 0.123$ performs better than the other two variables. In this jet substructure variable, the quark jets are less well described compared to the gluon jets in Figure 6.14a. This might be because of the larger uncertainties due to lower statistics for Z+jet events.
Overall, Z+jet events were well described in all jet substructure variables. They were expected to be better described as quark jet shapes were already well constraint in experiments at the Large Electron-Positron Collider (LEP) while gluon jet shapes are less well understood.
Identical to dijet event jets, the simulation of jet substructure performs better with $\alpha_S = 0.123$ compared to the default value for Z+jet events. However, as this was not a fine-tuned value, it can again only be said, that an $\alpha_S > 0.118$ performs the best. Additionally, varying only α_S is not enough to simulate the data well in all p_T-bins and substructure variables.

6.5. Underlying Event Tune

In Figures 6.15 and 6.16, underlying event measurements are presented. The plots in Figure 6.15 show the number of charged hadrons per η-bin and the plots in Figure 6.16 the energy density versus the p_T of the leading track in the transMin region. Further underlying event plots can be found in Appendices A.7 and A.8. In all four plots,



(a) Pythia8 **(b)** Herwig7

Figure 6.16.: Energy density versus the p_T of the leading track in the transMin region with α_S variation

the data is in black and the α_S-values 0.113, 0.118 and 0.123 are the blue, green and yellow line, respectively. Additionally, two more α_S values are displayed in the plots to increase the visibility of the effects of α_S variation: $\alpha_S = 0.100$ in red and $\alpha_S = 0.136$ in violet. In the bottom plot, the ratio *MC/data*, where MC (Monte Carlo generator) is either Pythia8 or Herwig7, is displayed for all α_S-values.

In Figure 6.15a the default value $\alpha_S = 0.118$ describes the data the best. However, it is not discernible, which α_S value performs the best for Herwig7 in Figure 6.15b. For both generators, a trend is visible when varying α_S: the larger α_S, the larger the number of charged hadrons per η-bin and vice versa. However, this trend is less pronounced for Herwig7. In both plots in Figure 6.16, this trend is visible as well: the lower the α_S, the lower the energy density per p_T-bin and vice versa. Again, this trend is more pronounced for Herwig7. In both of these plots, it is not apparent, which α_S value performs the best.

Overall, these results again suggest that Herwig7 is better suited to model jet substructure as it not only performs better than Pythia8 in some jet substructure and p_T-bins but also shows less of a trend in the underlying event measurements when varying α_S. However, it is again important to note, that the α_S variable varied in each of the generators have differences in their effects: for Pythia8, it only affects the final state radiation, while it affects every step of the generation process for Herwig7.

Chapter 7.

Conclusion

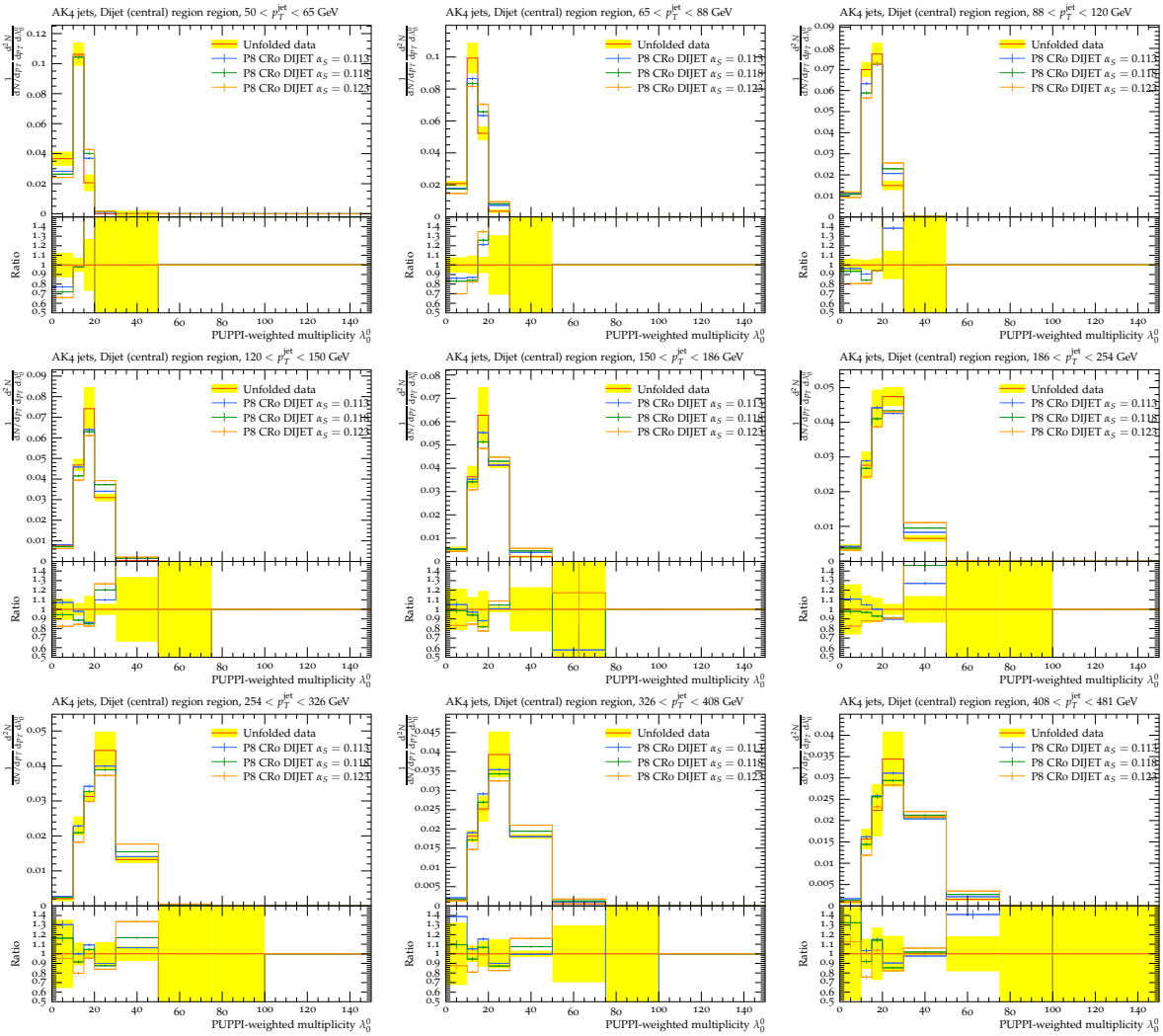
The effects of varying α_S on the substructure simulation for Pythia8 and Herwig7 were presented. A higher α_S leads to more interactions and, therefore, more particles with less energy per particle. This is directly visible in the multiplicity and $(p_T^D)^2$. The Colour Reconnection models for Pythia8 all perform similarly well when simulating the jet substructure observables. All three CR modes are interesting for further studies involving the fine-tuning of α_S combined with other variables. The MC generators Pythia8 and Herwig7 were compared in their ability to simulate the jet substructure observables. Herwig7 performs at least similarly well and in some jet substructure and p_T-bins better compared to Pythia8. This might imply, that Herwig7 is better suited to model jet substructure. The modelling of quark-jets was well described in all jet substructure variables, while gluon-jets were only well described in LHA, width and thrust. Quark-jets were expected to be better described as quark-jet shapes were already well constraint in experiments at the Large Electron-Positron Collider (LEP) while gluon-jet shapes are less well understood. The effects of varying α_S on the underlying event tune were presented. Herwig7 shows less of a trend in the underlying event measurements when varying α_S. These results again suggest that Herwig7 is better suited to model jet substructure.
Overall, for dijet event jets, Herwig7 and Pythia8 model LHA, width and thrust well, but multiplicity and $(p_T^D)^2$ are difficult to describe. While for Z+jet event jets, all jet substructure observables were well described.
The simulation of quark- and gluon-jet jet substructure observables performs better with $\alpha_S = 0.123$ compared to the default value for both generators. However, as this was not a fine-tuned value, it can only be said, that an $\alpha_S > 0.118$ performs the best. Additionally, it is not possible to simulate the data well in all p_T-bins and substructure variables when varying only α_S.

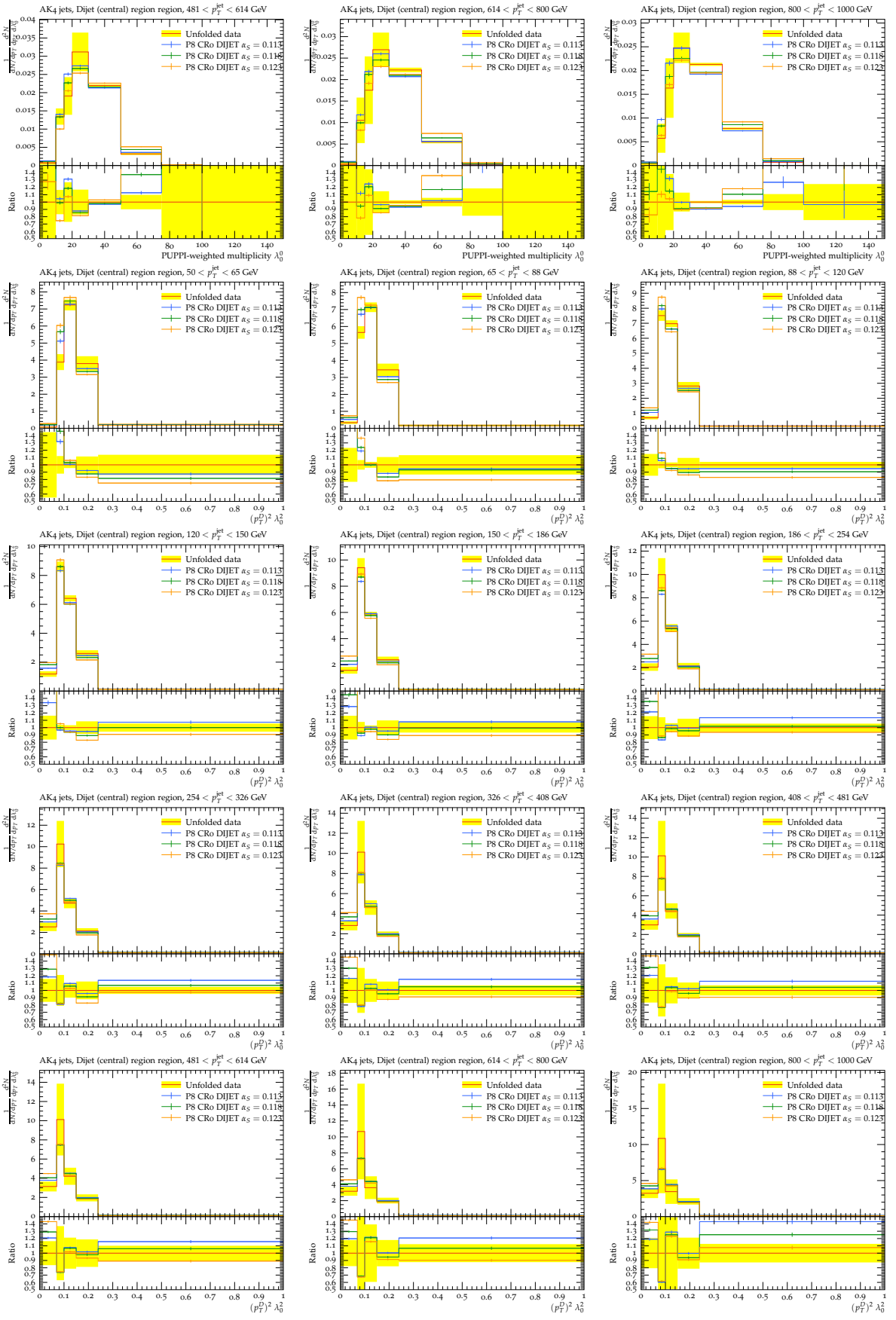
Herwig7 might be better suited to model jet substructure. However, it is once again important to note, that the α_S variable varied in each of the generators have differences in their effects: for Pythia8, it only affects the final state radiation, while it affects every step of the generation process for Herwig7.
In further studies, the α_S variable can be fine-tuned, especially in combination with further variables. Additionally, the substructure variables can be examined for AK8-jets and groomed jets and when only considering charged hadrons. These are explored in Reference [4]. Another important addition is the calculation of χ^2 for the agreement of the simulations to the data to have a numerical value to verify the results from the plots.

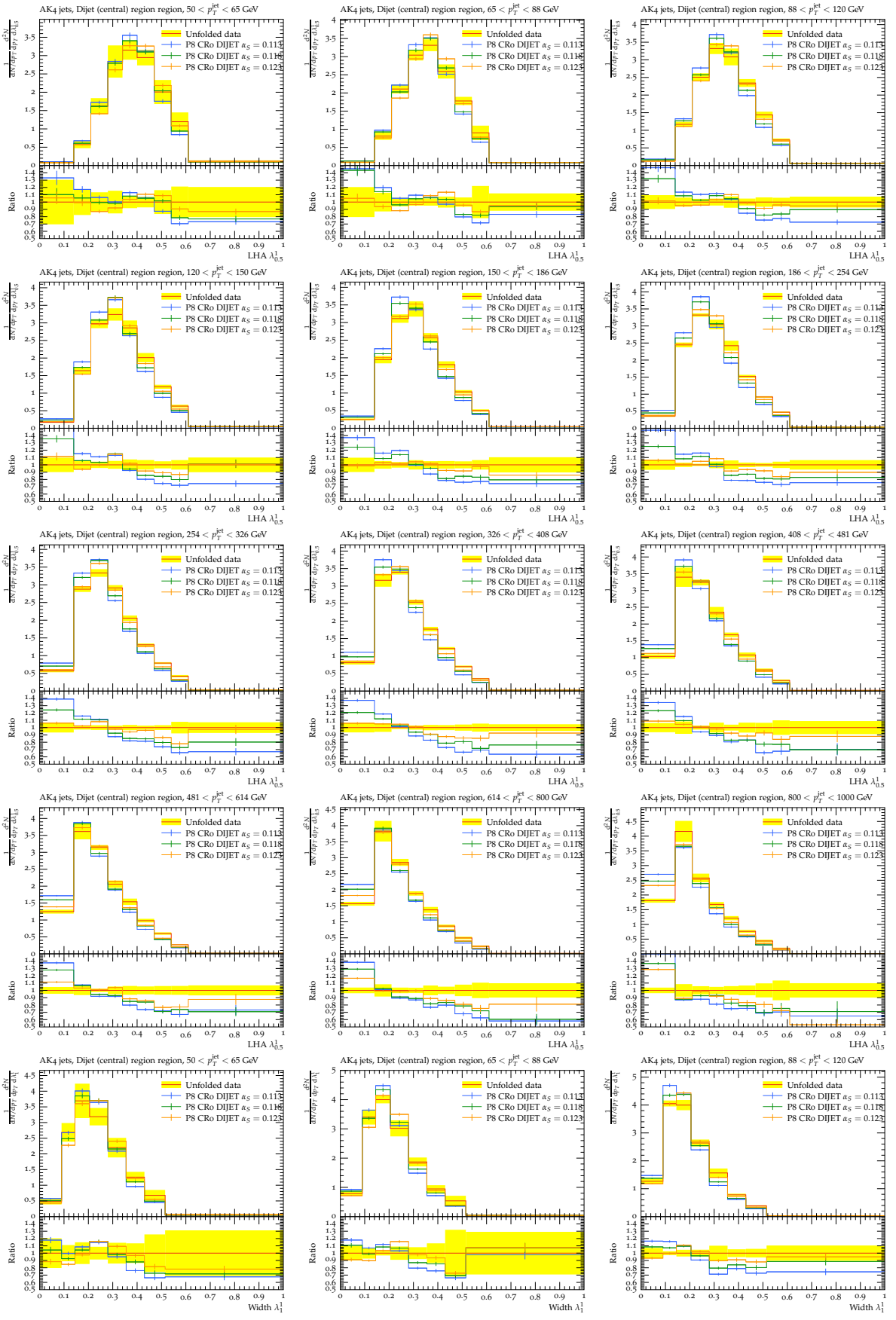
Appendix A.

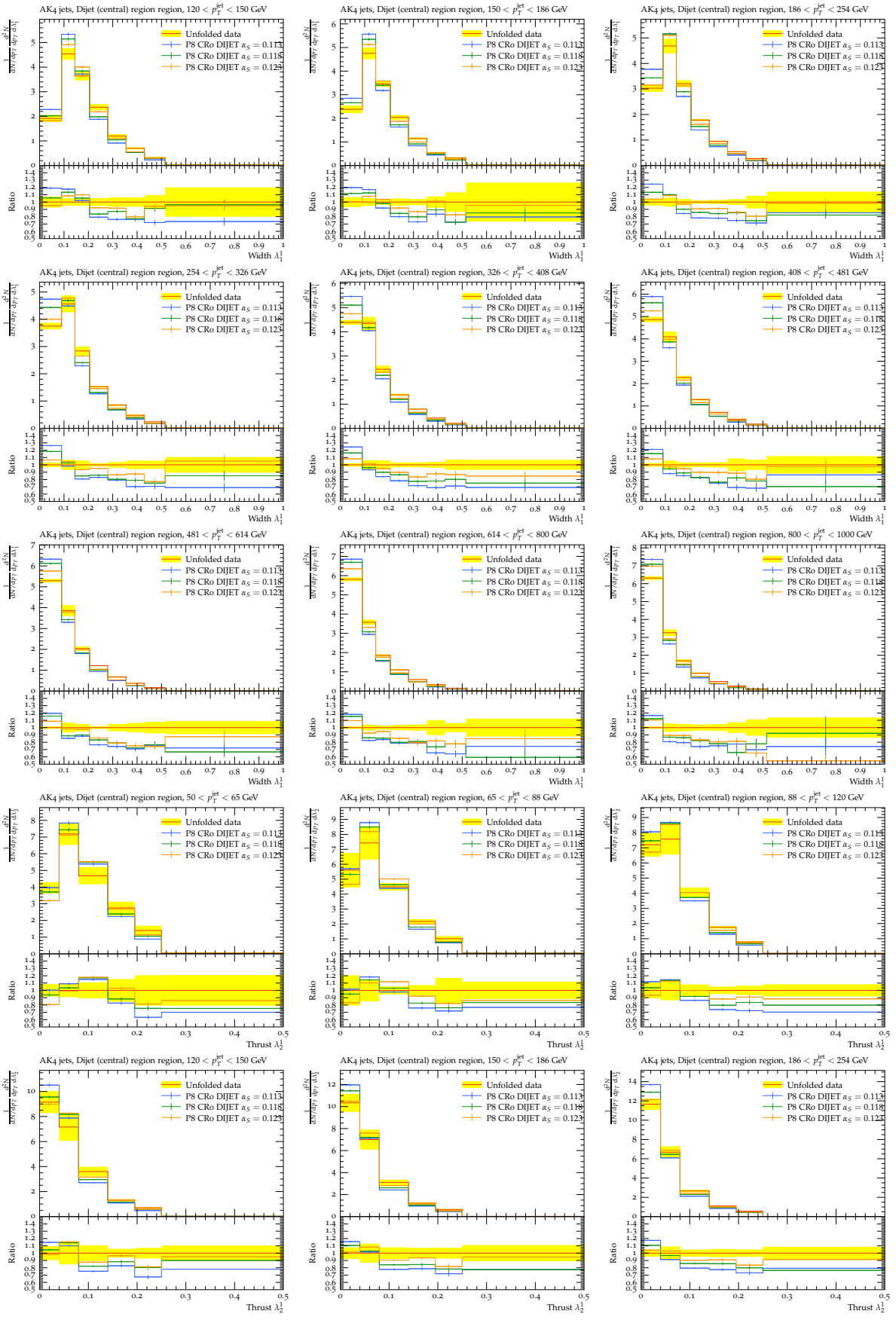
Additional Plots

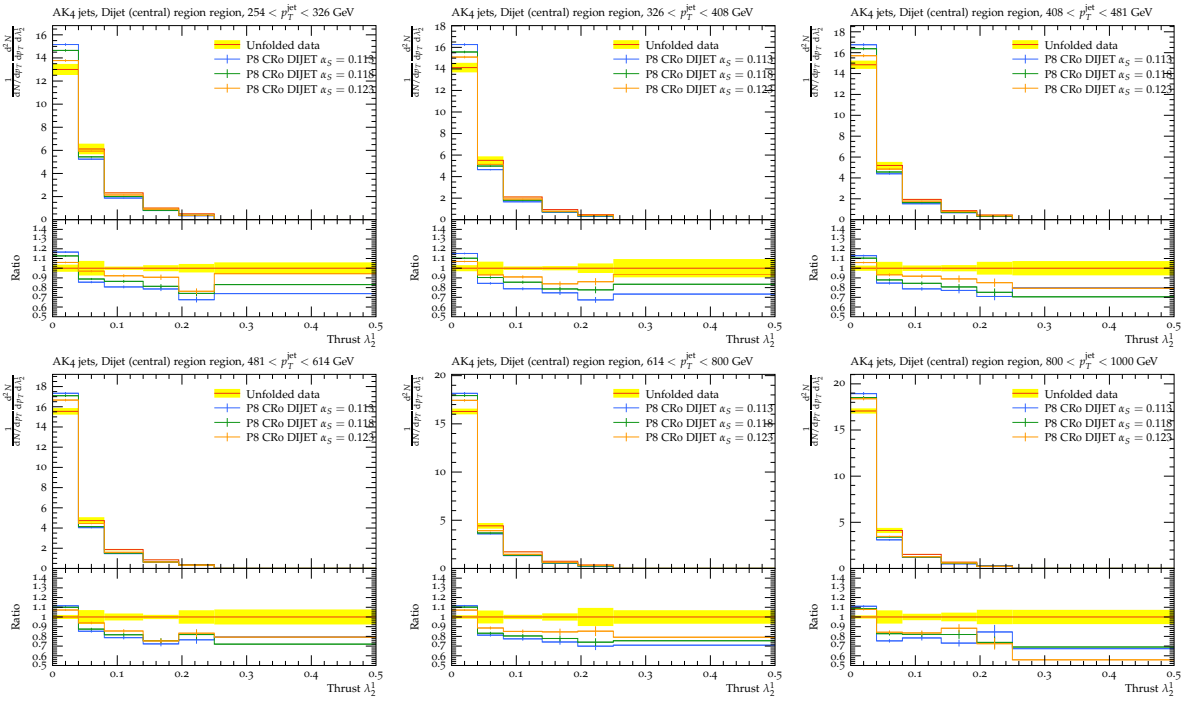
A.1. Pythia8 CR0 Dijet Plots



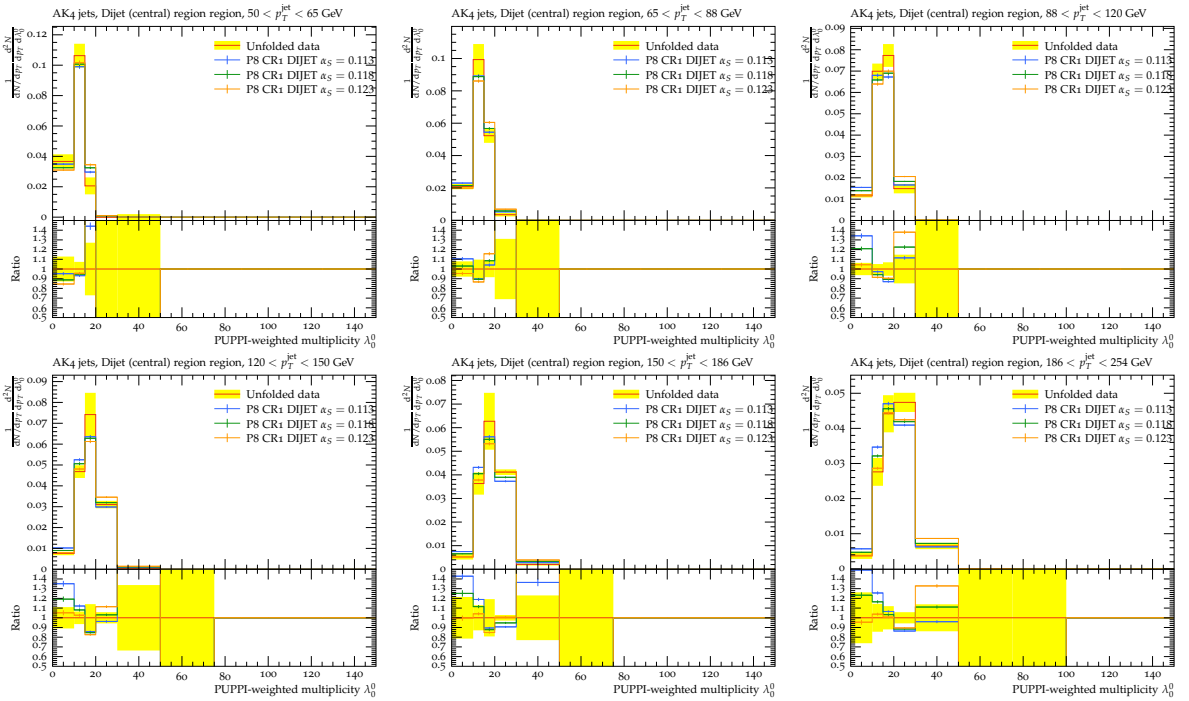

AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
Unfolded data
P8 CRo DIJET $\alpha_S = 0.113$
P8 CRo DIJET $\alpha_S = 0.118$
P8 CRo DIJET $\alpha_S = 0.123$
PUPPI-weighted multiplicity λ_0^0
$(p_T^D)^2$ λ_0^2
Ratio

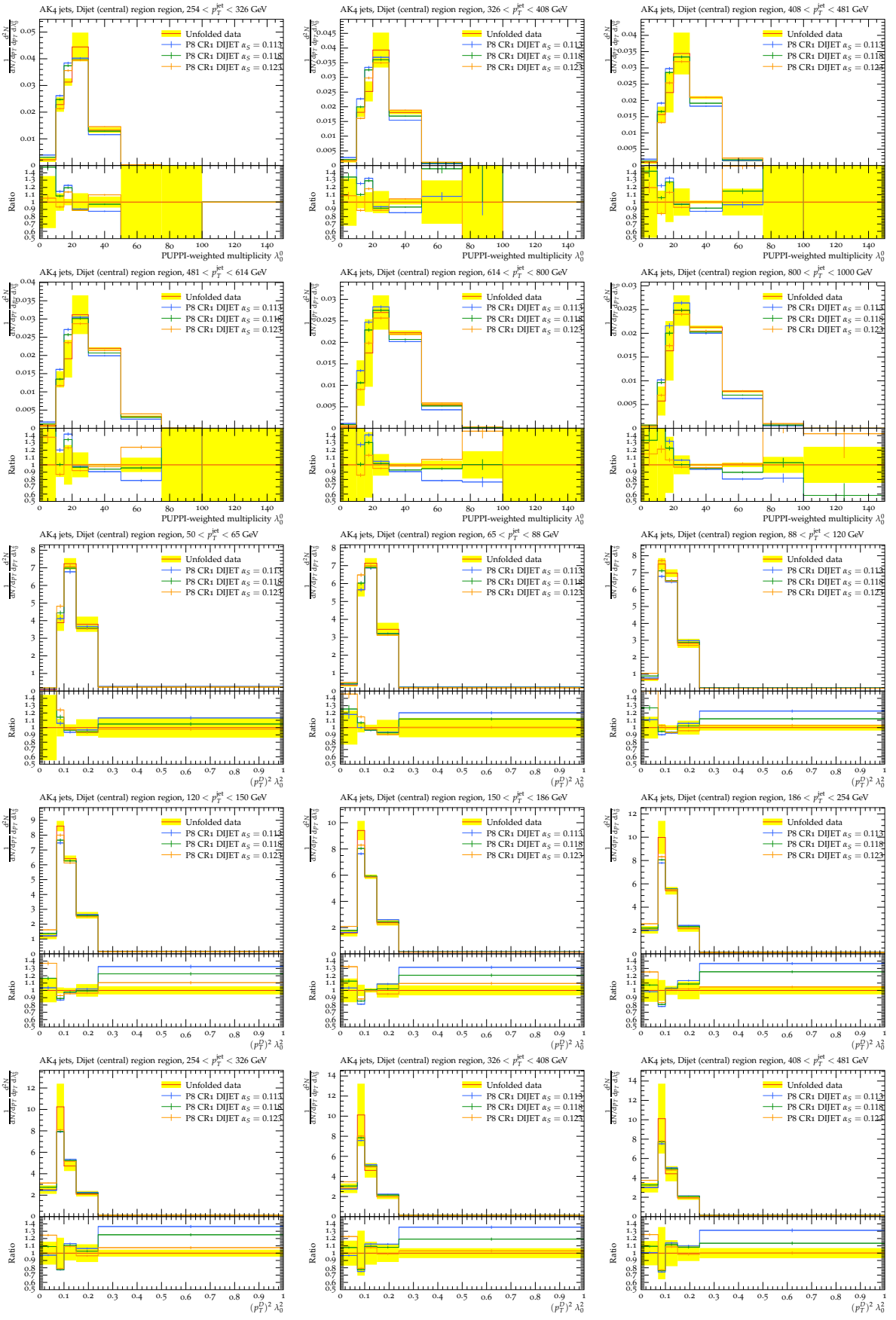

AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
Unfolded data
P8 CRo DIJET $\alpha_S = 0.113$
P8 CRo DIJET $\alpha_S = 0.118$
P8 CRo DIJET $\alpha_S = 0.123$
Ratio
LHA $\lambda^1_{0.5}$
Width λ^1_1

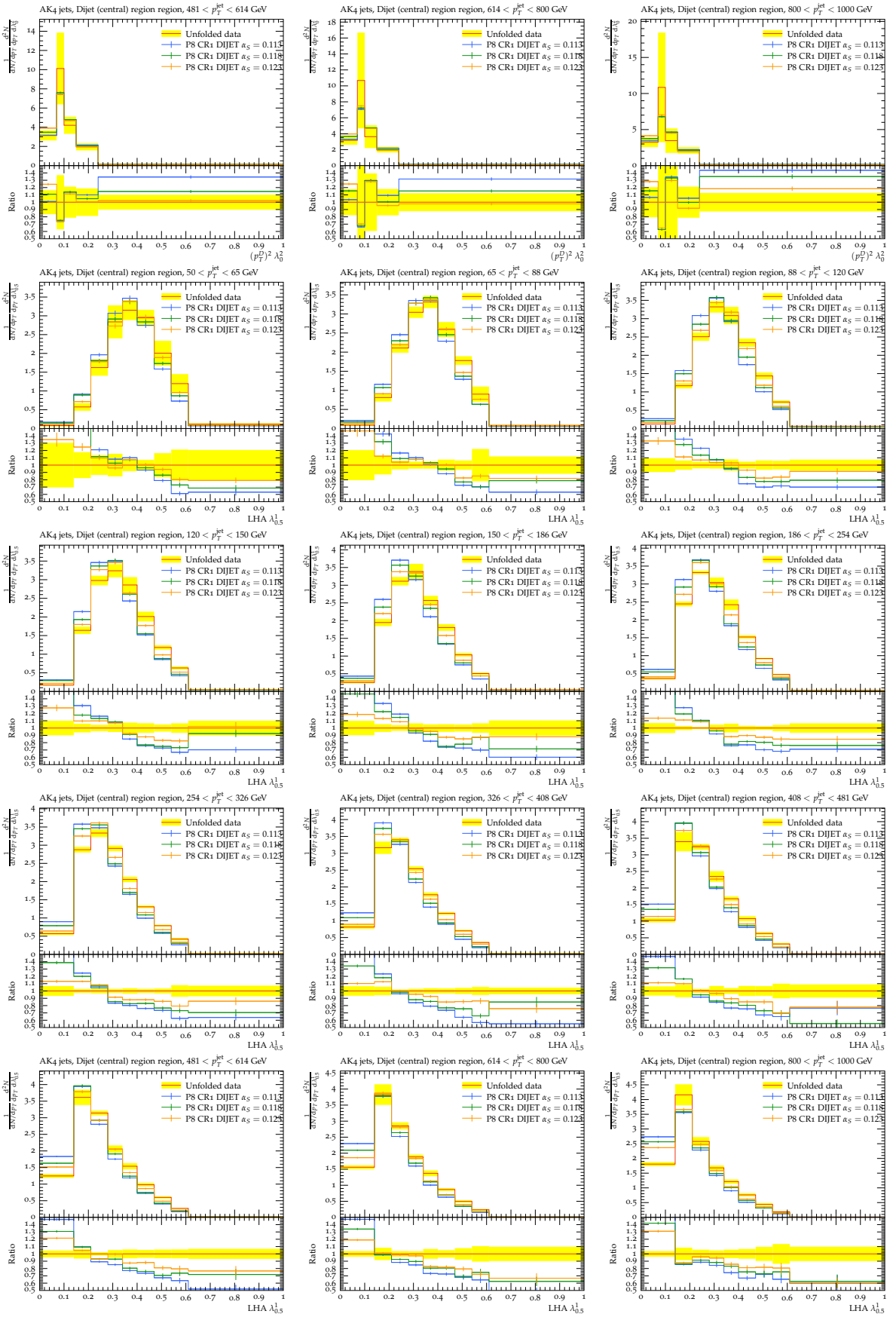

AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
Unfolded data
P8 CRo DIJET $\alpha_S = 0.113$
P8 CRo DIJET $\alpha_S = 0.118$
P8 CRo DIJET $\alpha_S = 0.123$
Ratio
Width λ_1^1
Thrust λ_2^1

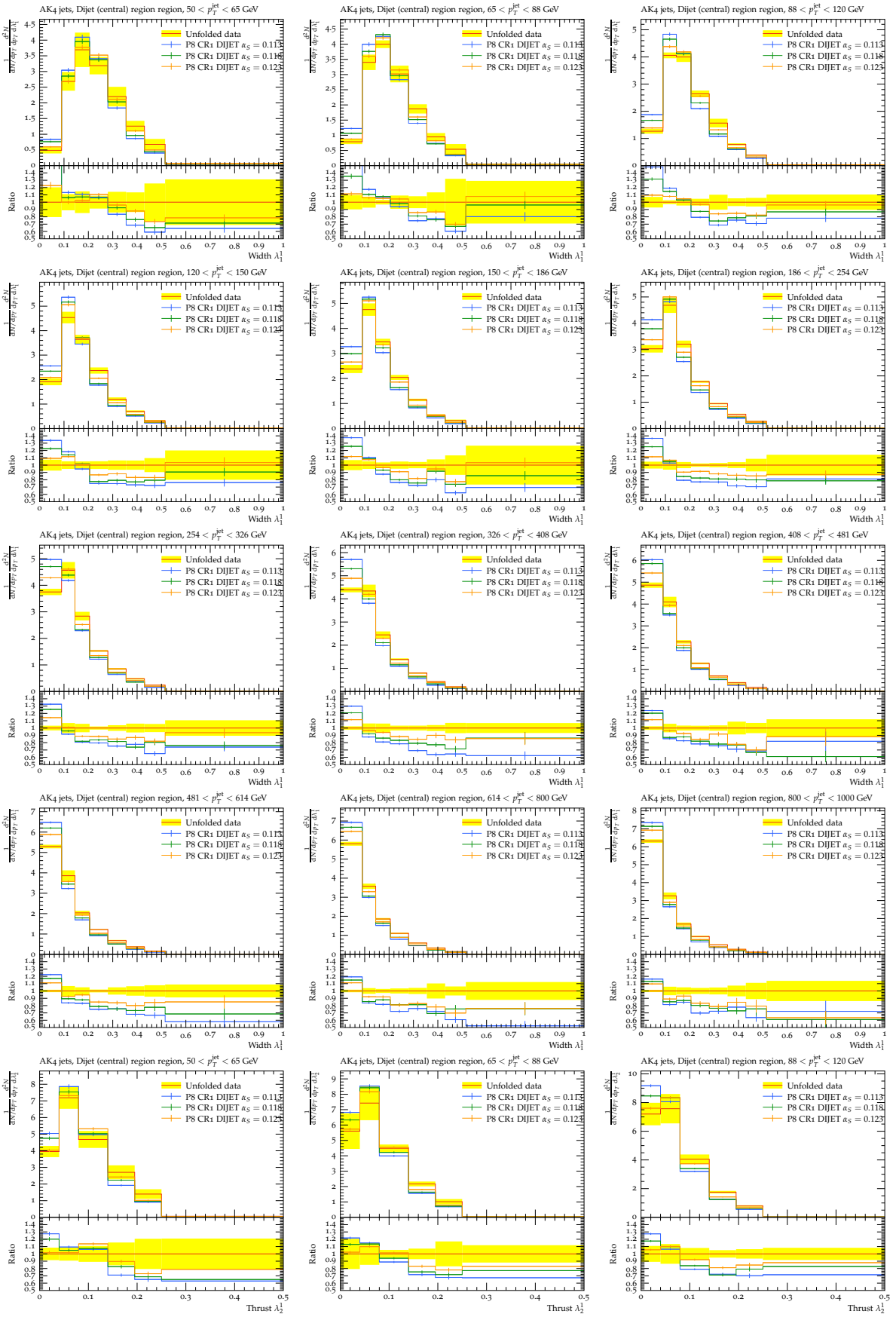


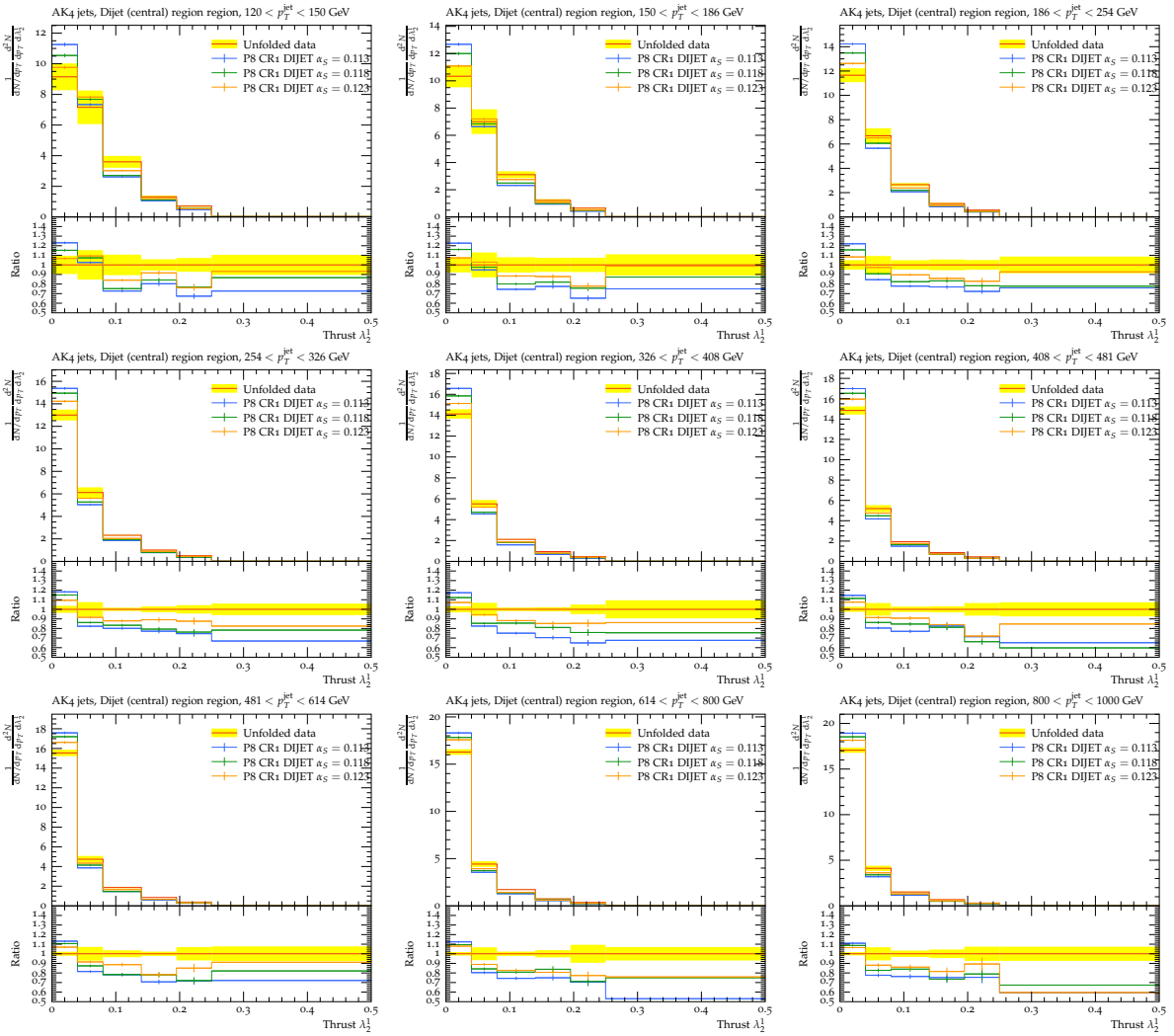
A.2. Pythia8 CR1 Dijet Plots



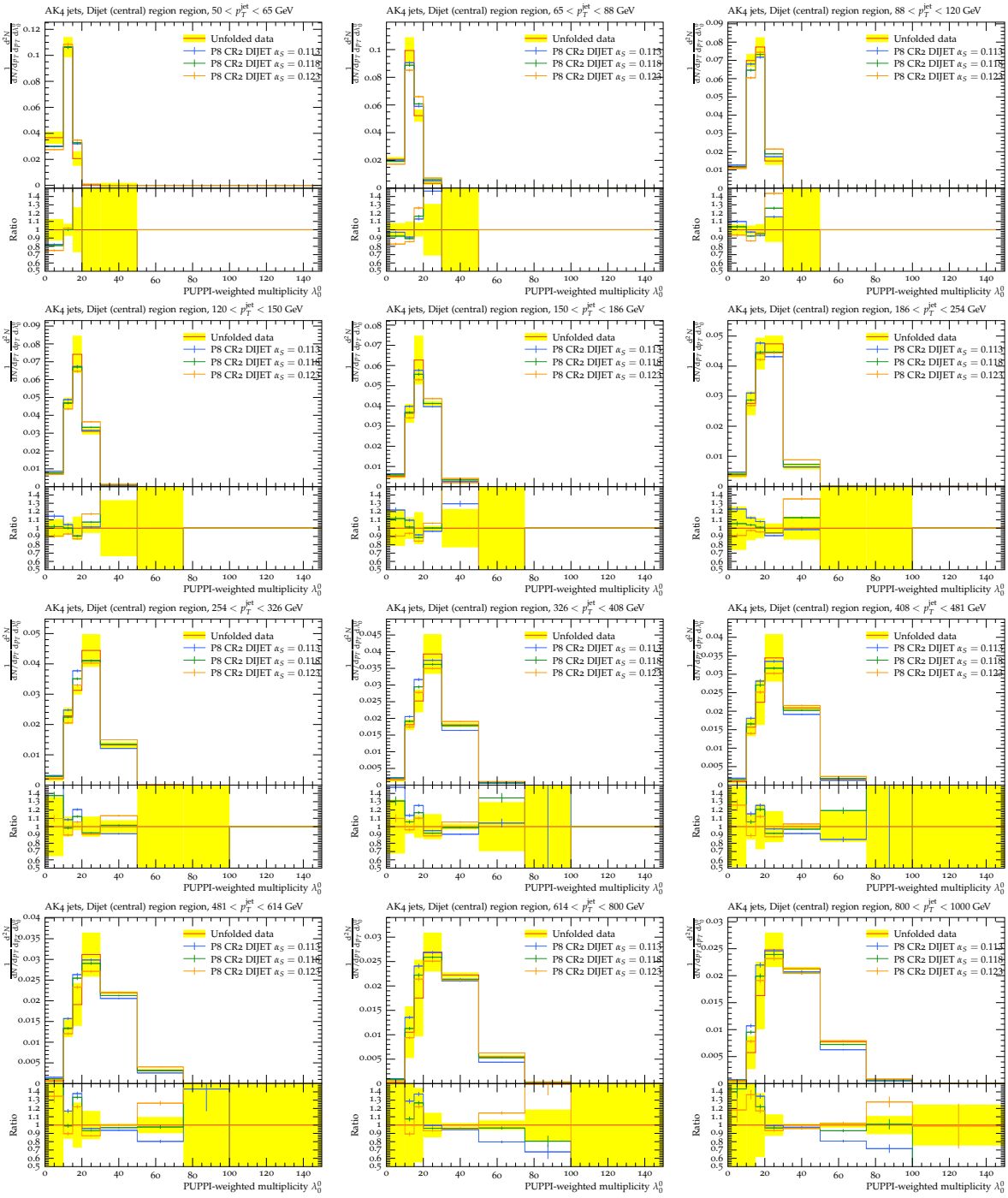

AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
Unfolded data
P8 CR1 DIJET $\alpha_S = 0.113$
P8 CR1 DIJET $\alpha_S = 0.118$
P8 CR1 DIJET $\alpha_S = 0.123$
Ratio
PUPPI-weighted multiplicity λ_0^0
$(p_T^D)^2$ λ_0^2

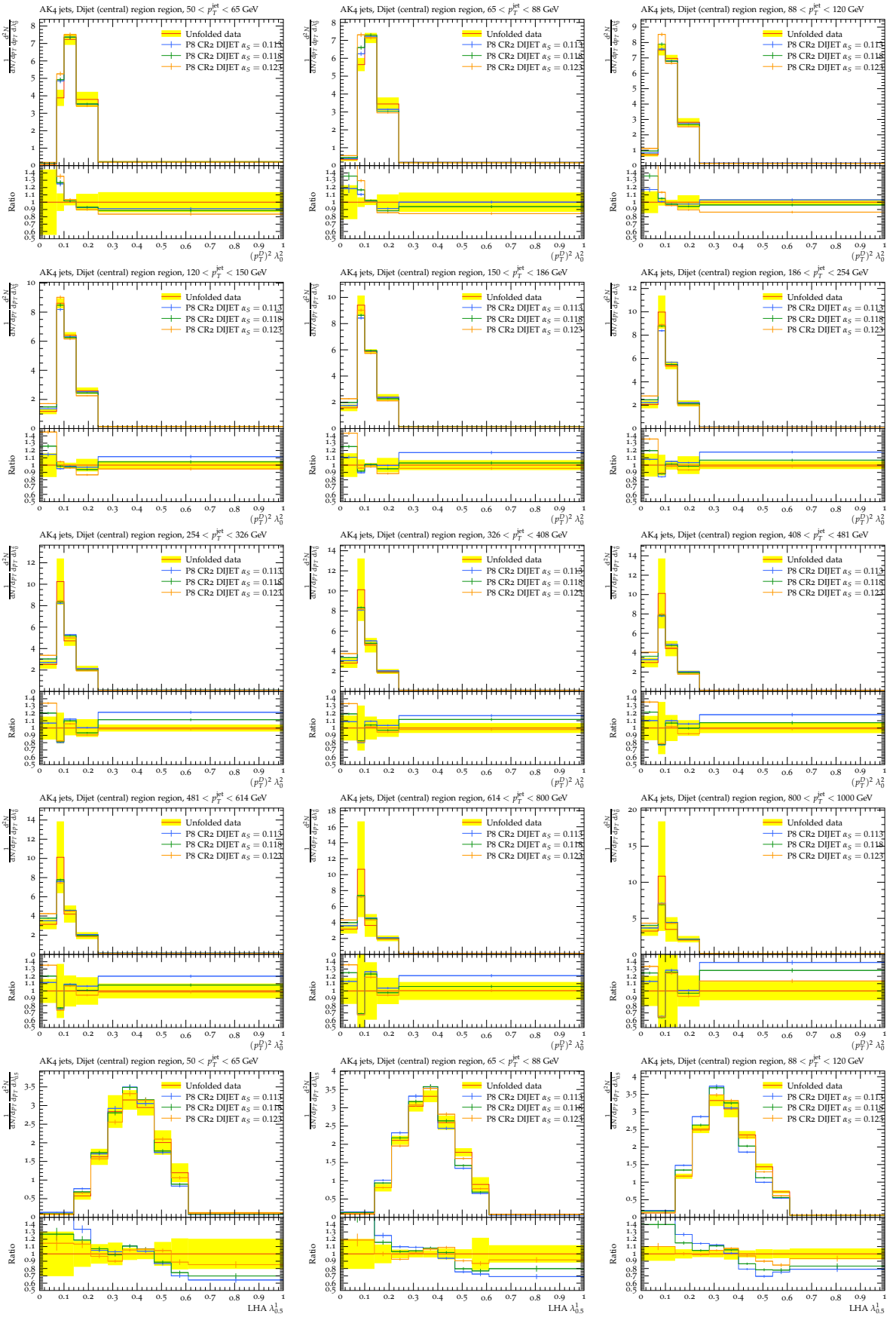

AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
Unfolded data
P8 CR1 DIJET $\alpha_S = 0.113$
P8 CR1 DIJET $\alpha_S = 0.118$
P8 CR1 DIJET $\alpha_S = 0.123$
Ratio
$(p_T^D)^2\ \lambda_0^2$
LHA $\lambda_{0.5}^1$

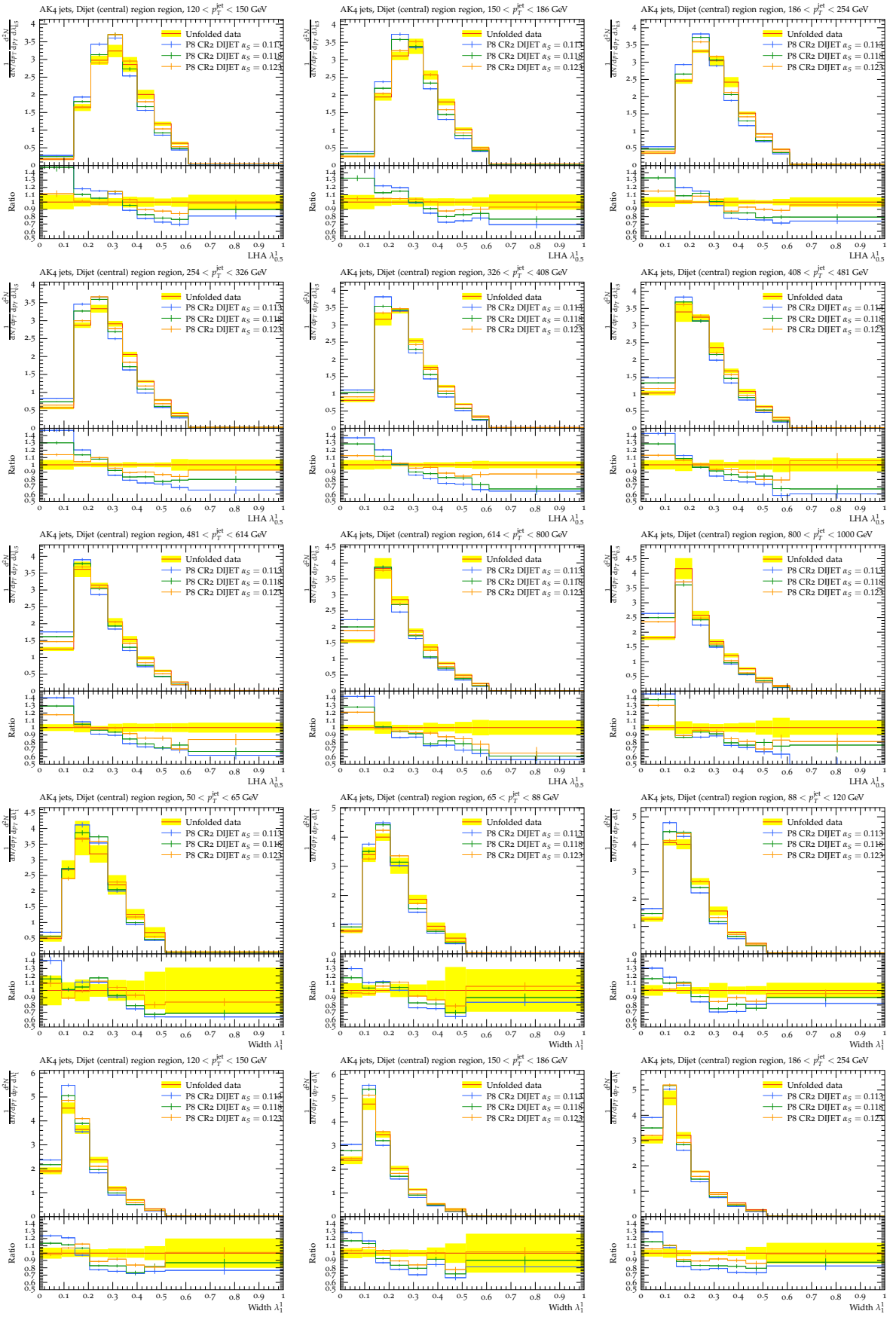

AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
Unfolded data
P8 CR1 DIJET $\alpha_S = 0.113$
P8 CR1 DIJET $\alpha_S = 0.118$
P8 CR1 DIJET $\alpha_S = 0.123$
Ratio
Width λ_1^1
Thrust λ_2^1

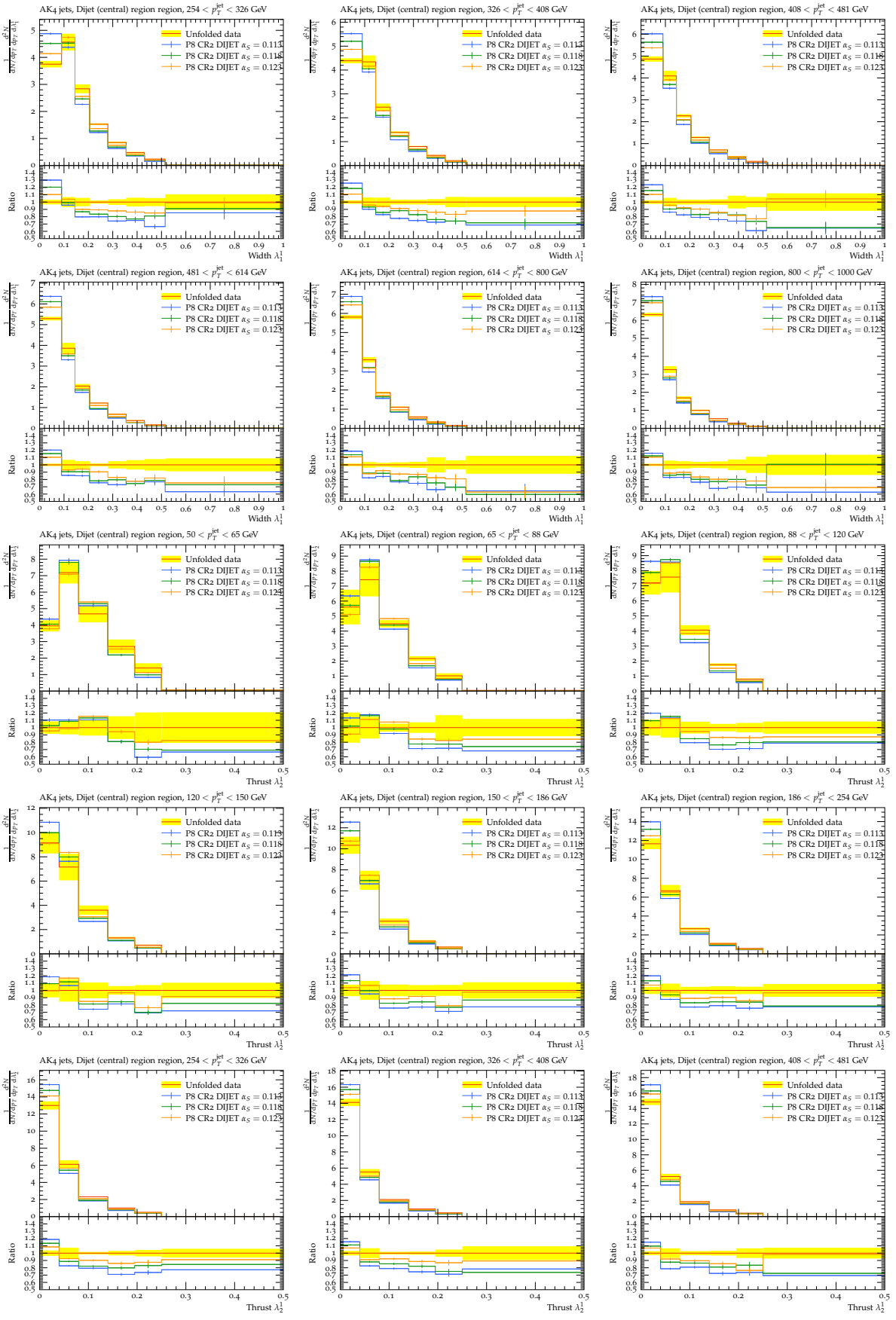

AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
Unfolded data
P8 CR1 DIJET $\alpha_S = 0.113$
P8 CR1 DIJET $\alpha_S = 0.118$
P8 CR1 DIJET $\alpha_S = 0.123$
Ratio
Thrust λ_2^1

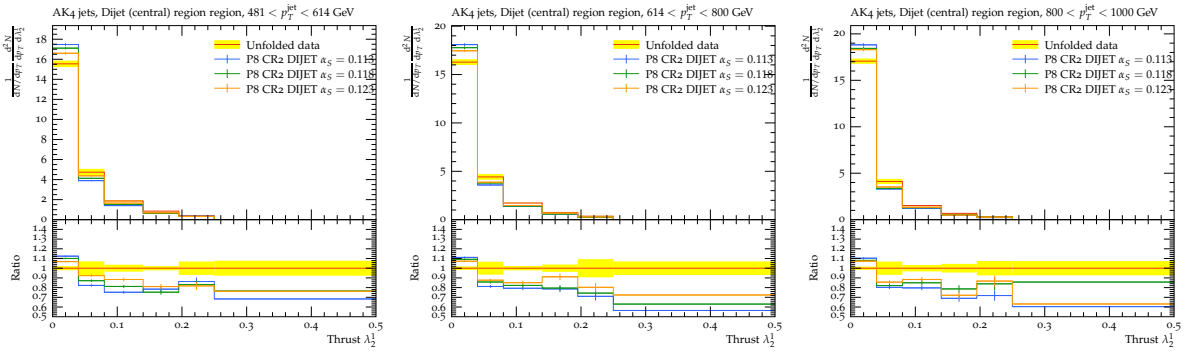
A.3. Pythia8 CR2 Dijet Plots



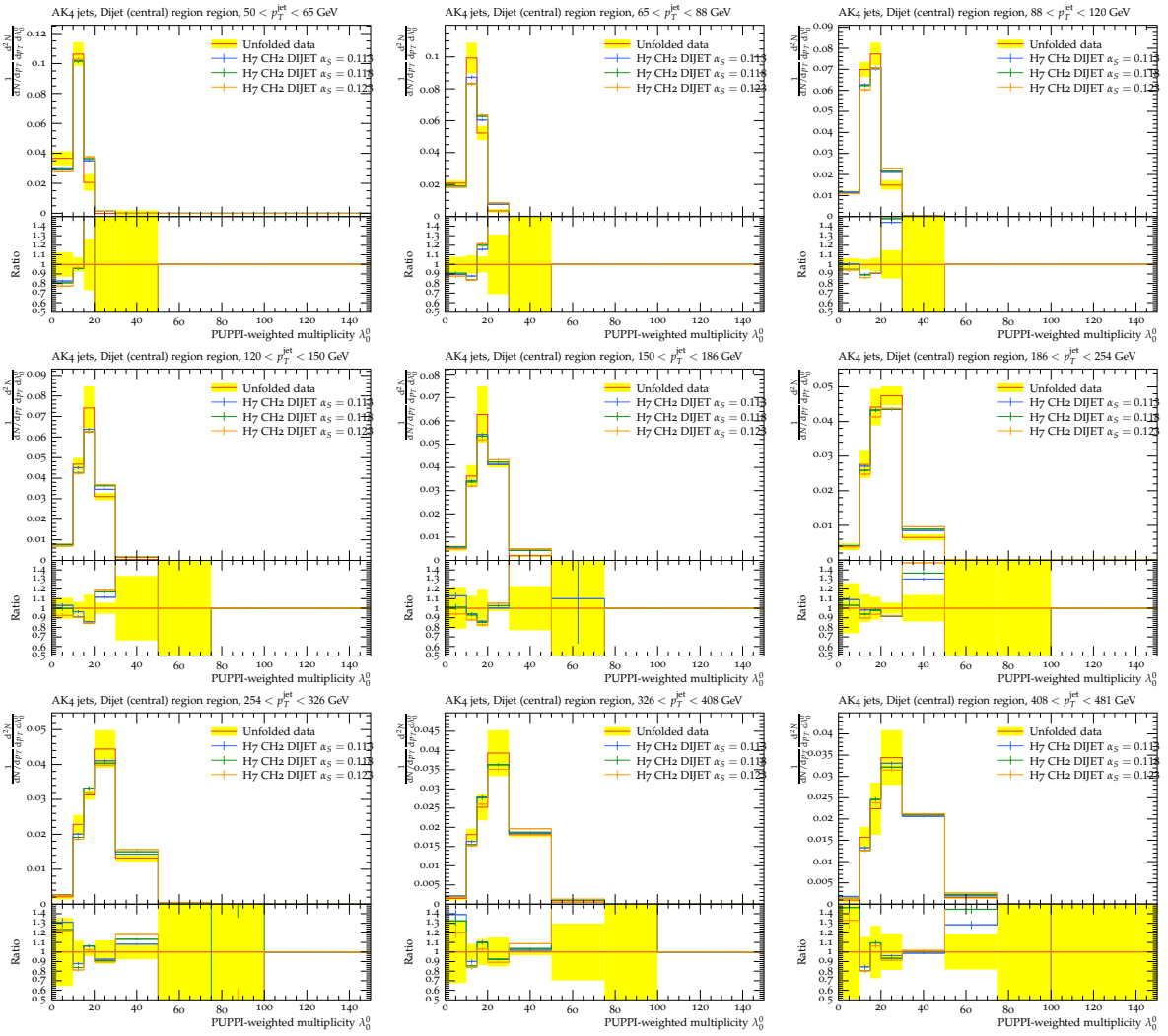

AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
Unfolded data
P8 CR2 DIJET $\alpha_S = 0.113$
P8 CR2 DIJET $\alpha_S = 0.118$
P8 CR2 DIJET $\alpha_S = 0.123$
Ratio
$(p_T^D)^2\ \lambda_0^2$
LHA $\lambda_{0.5}^1$

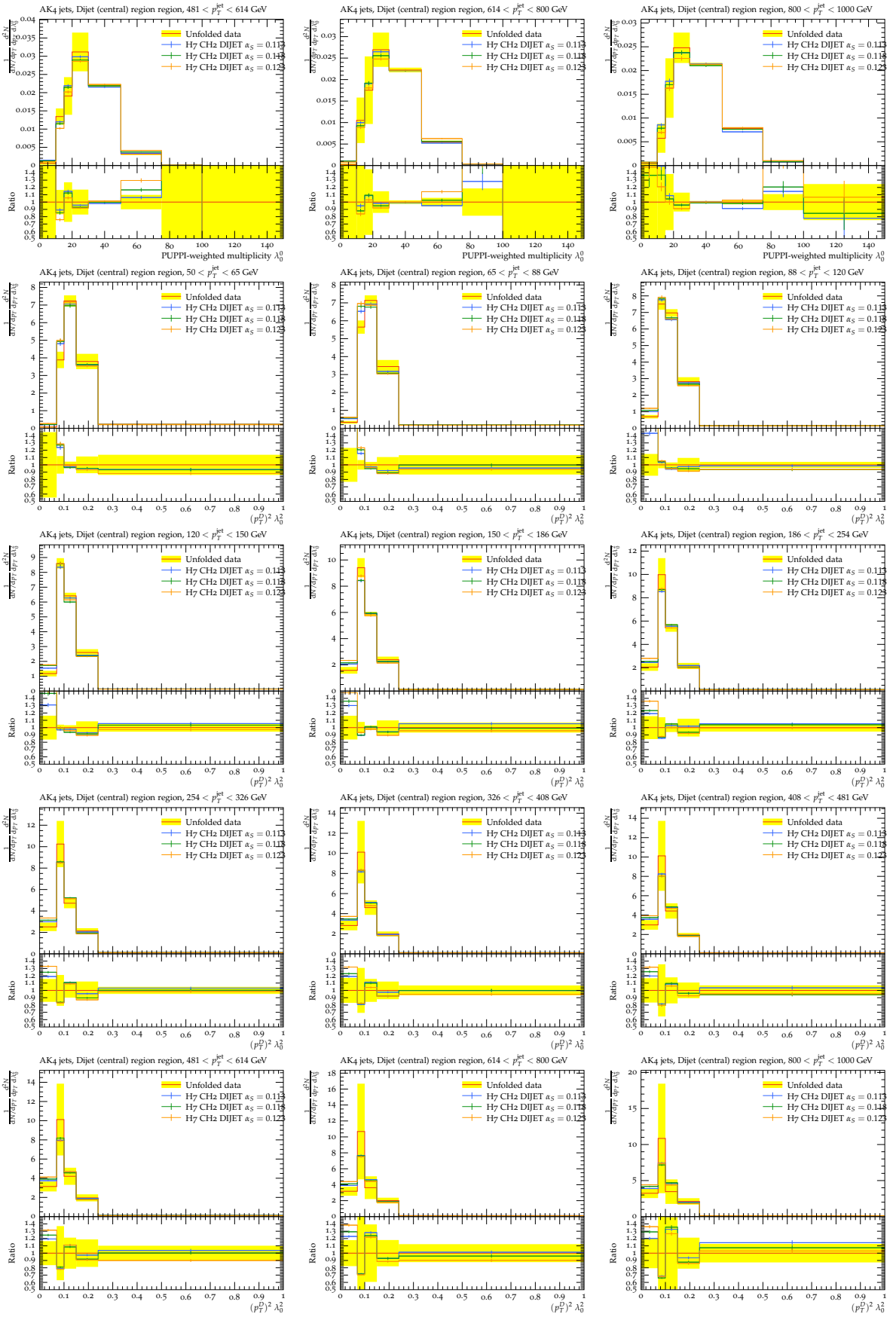

AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
Unfolded data
P8 CR2 DIJET $\alpha_S = 0.113$
P8 CR2 DIJET $\alpha_S = 0.118$
P8 CR2 DIJET $\alpha_S = 0.123$
Ratio
LHA $\lambda^1_{0.5}$
Width λ^1_1

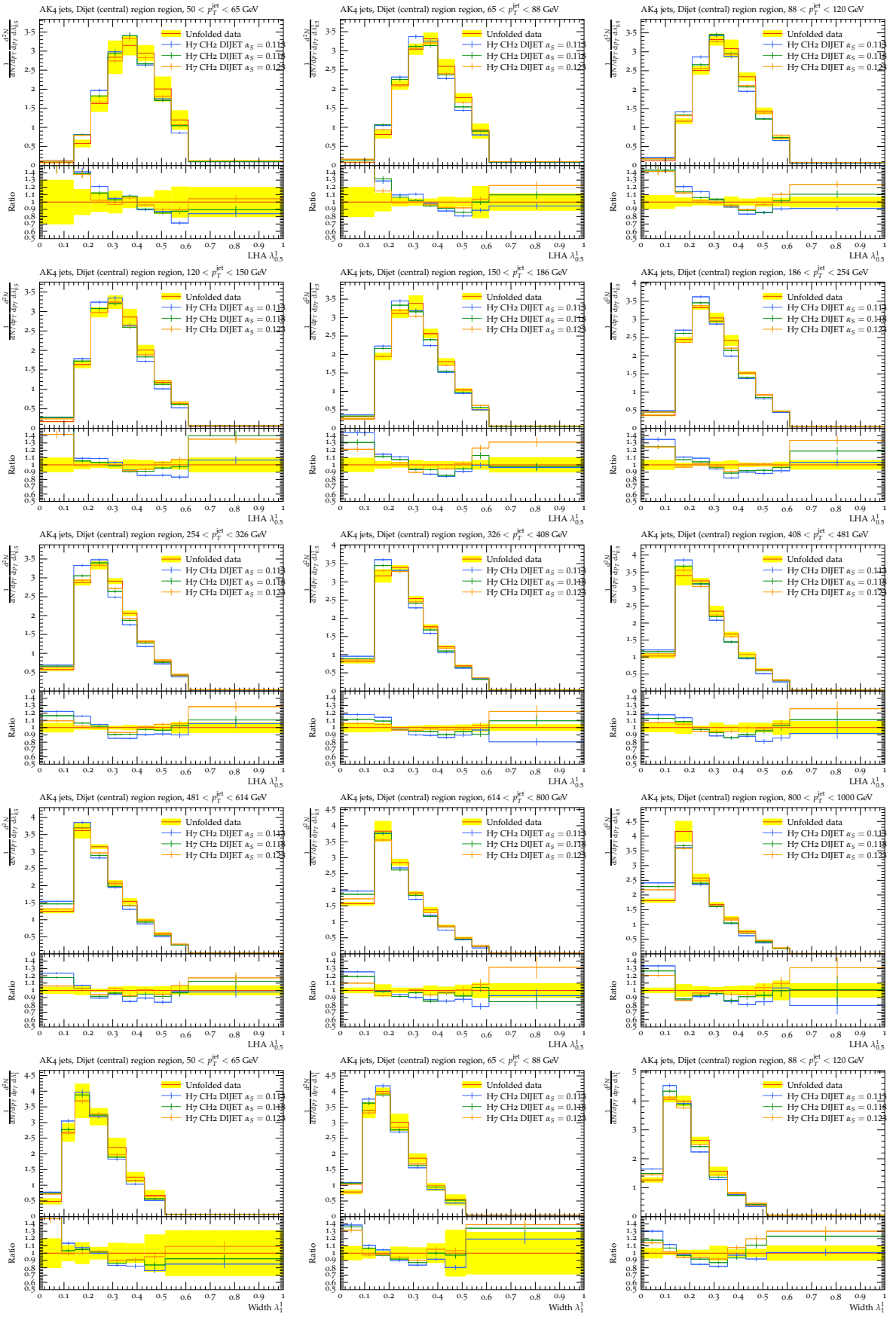

AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
Unfolded data
P8 CR2 DIJET $\alpha_S = 0.113$
P8 CR2 DIJET $\alpha_S = 0.118$
P8 CR2 DIJET $\alpha_S = 0.123$
Ratio
Width λ_1^1
Thrust λ_2^1

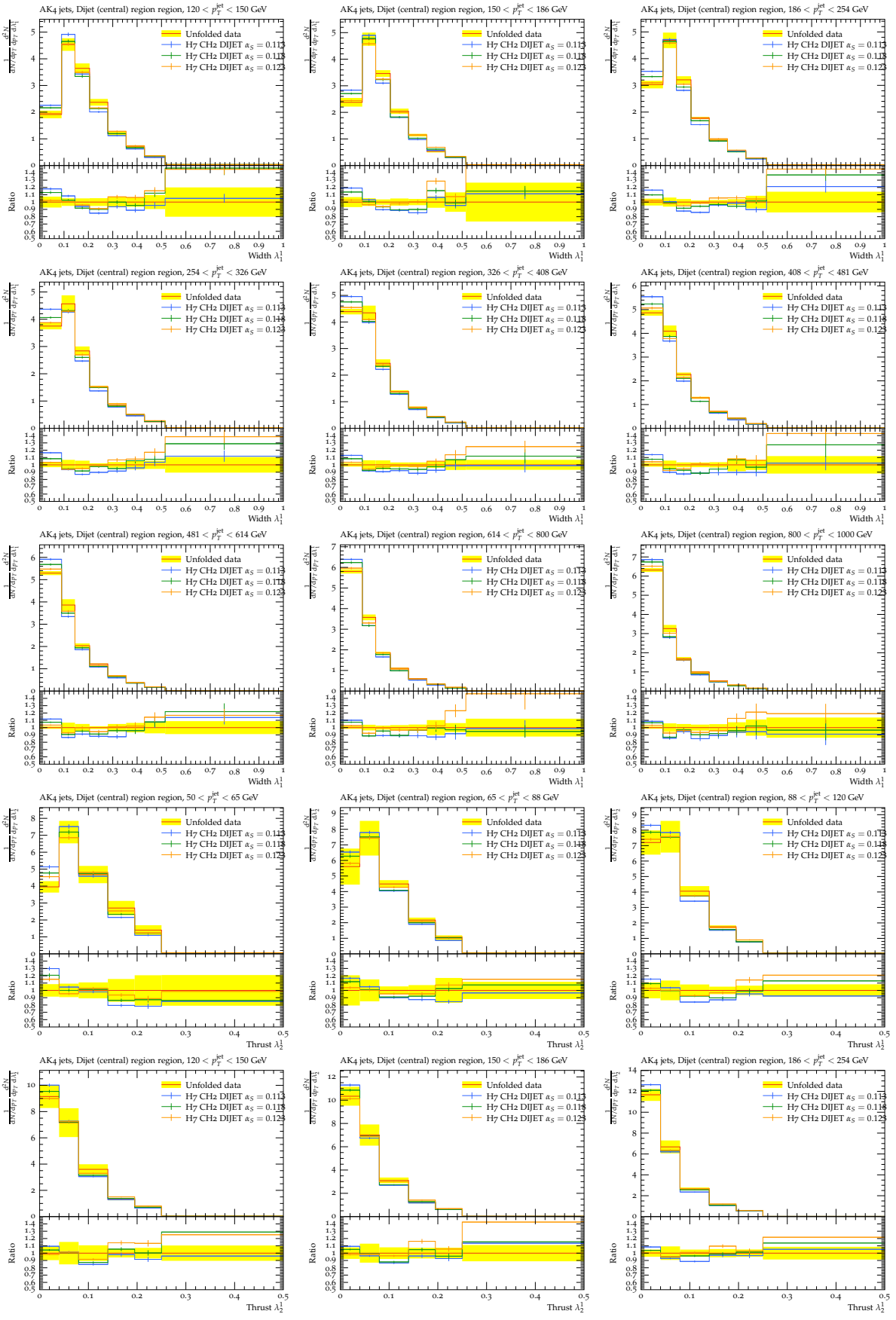


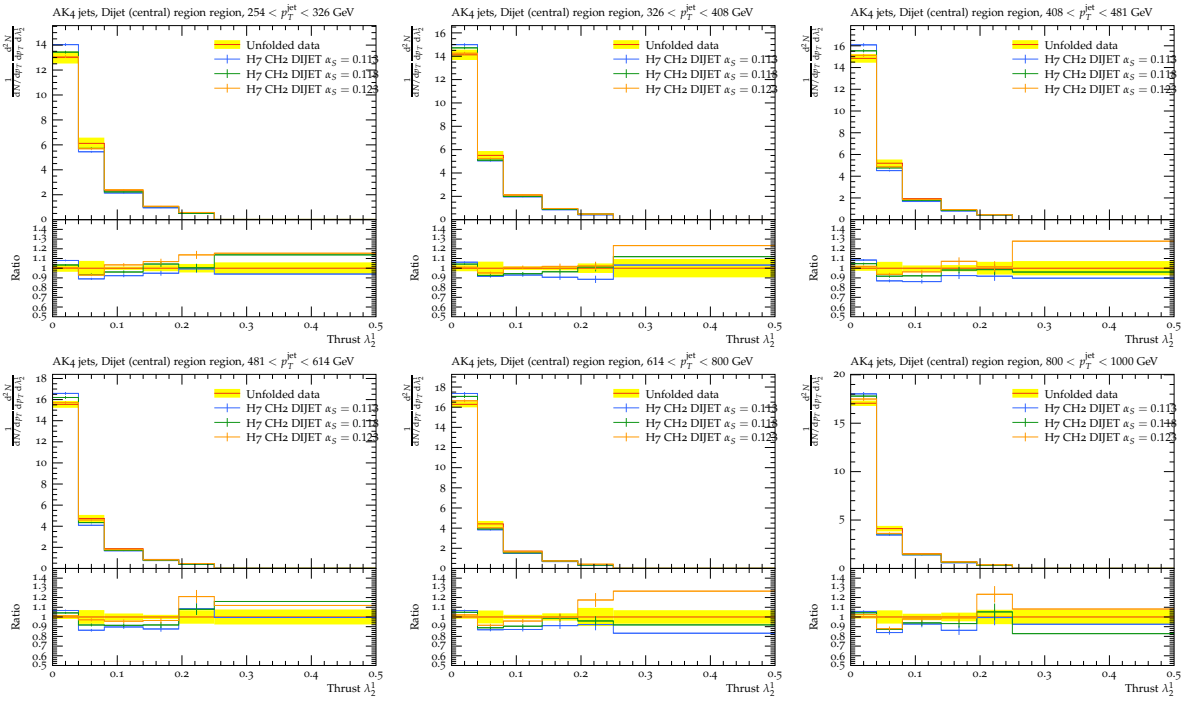
A.4. Herwig7 Dijet Plots



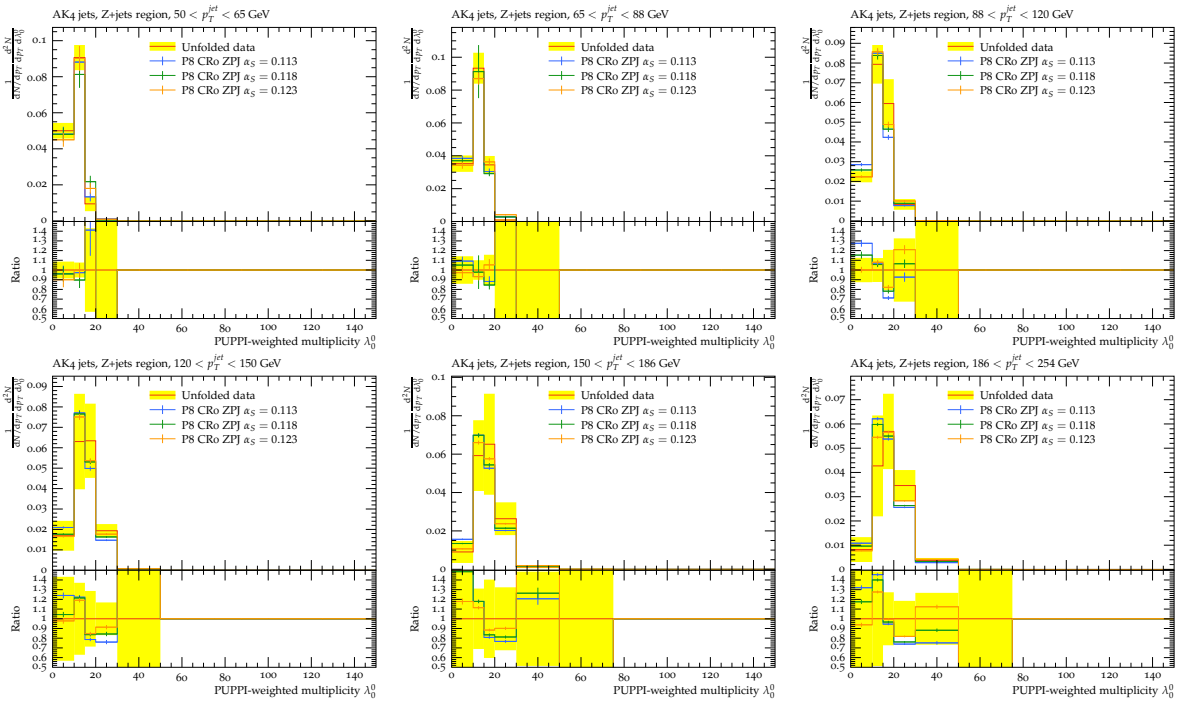

AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
Unfolded data
H7 CH2 DIJET $\alpha_S = 0.113$
H7 CH2 DIJET $\alpha_S = 0.118$
H7 CH2 DIJET $\alpha_S = 0.123$
PUPPI-weighted multiplicity λ_0^0
$(p_T^D)^2$ λ_0^2
Ratio

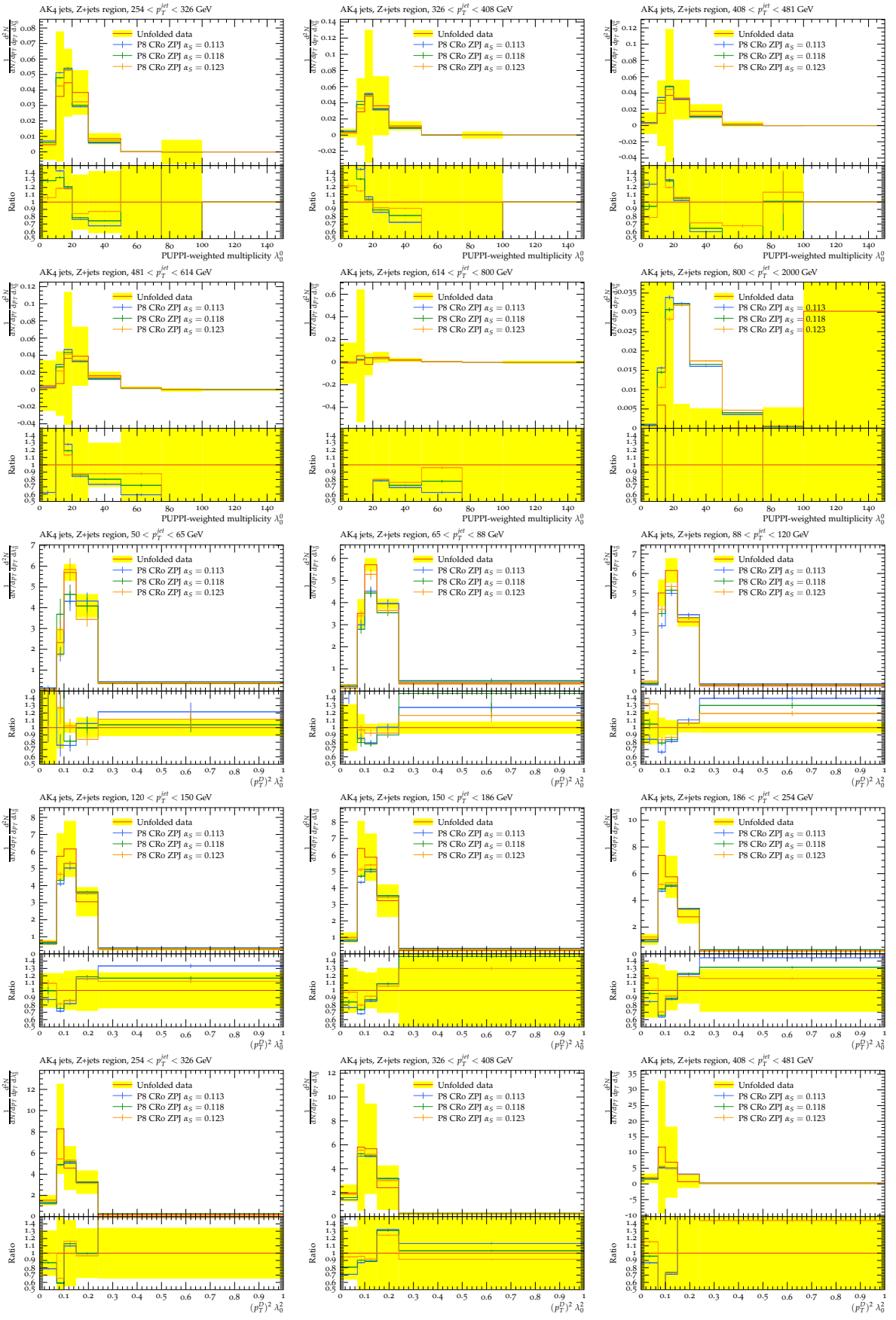

AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
Unfolded data
H7 CH2 DIJET $\alpha_S = 0.113$
H7 CH2 DIJET $\alpha_S = 0.118$
H7 CH2 DIJET $\alpha_S = 0.123$
Ratio
LHA $\lambda^1_{0.5}$
Width λ^1_1

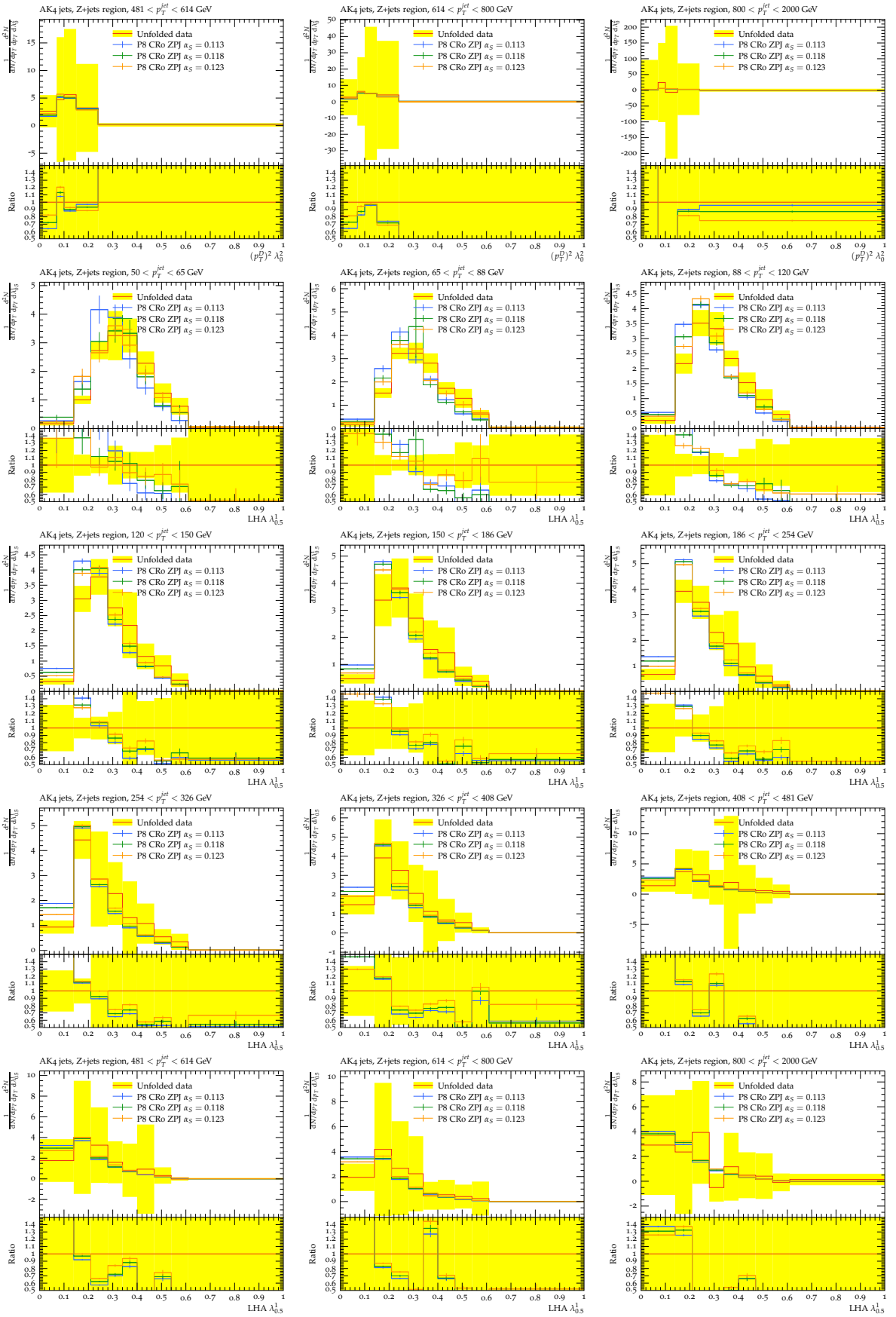

AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Dijet (central) region region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Dijet (central) region region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Dijet (central) region region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Dijet (central) region region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Dijet (central) region region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Dijet (central) region region, 800 < p_T^{jet} < 1000 GeV
AK4 jets, Dijet (central) region region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Dijet (central) region region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Dijet (central) region region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Dijet (central) region region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Dijet (central) region region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Dijet (central) region region, 186 < p_T^{jet} < 254 GeV
Unfolded data
H7 CH2 DIJET $\alpha_S = 0.113$
H7 CH2 DIJET $\alpha_S = 0.118$
H7 CH2 DIJET $\alpha_S = 0.123$
Ratio
Width λ_1^1
Thrust λ_2^1

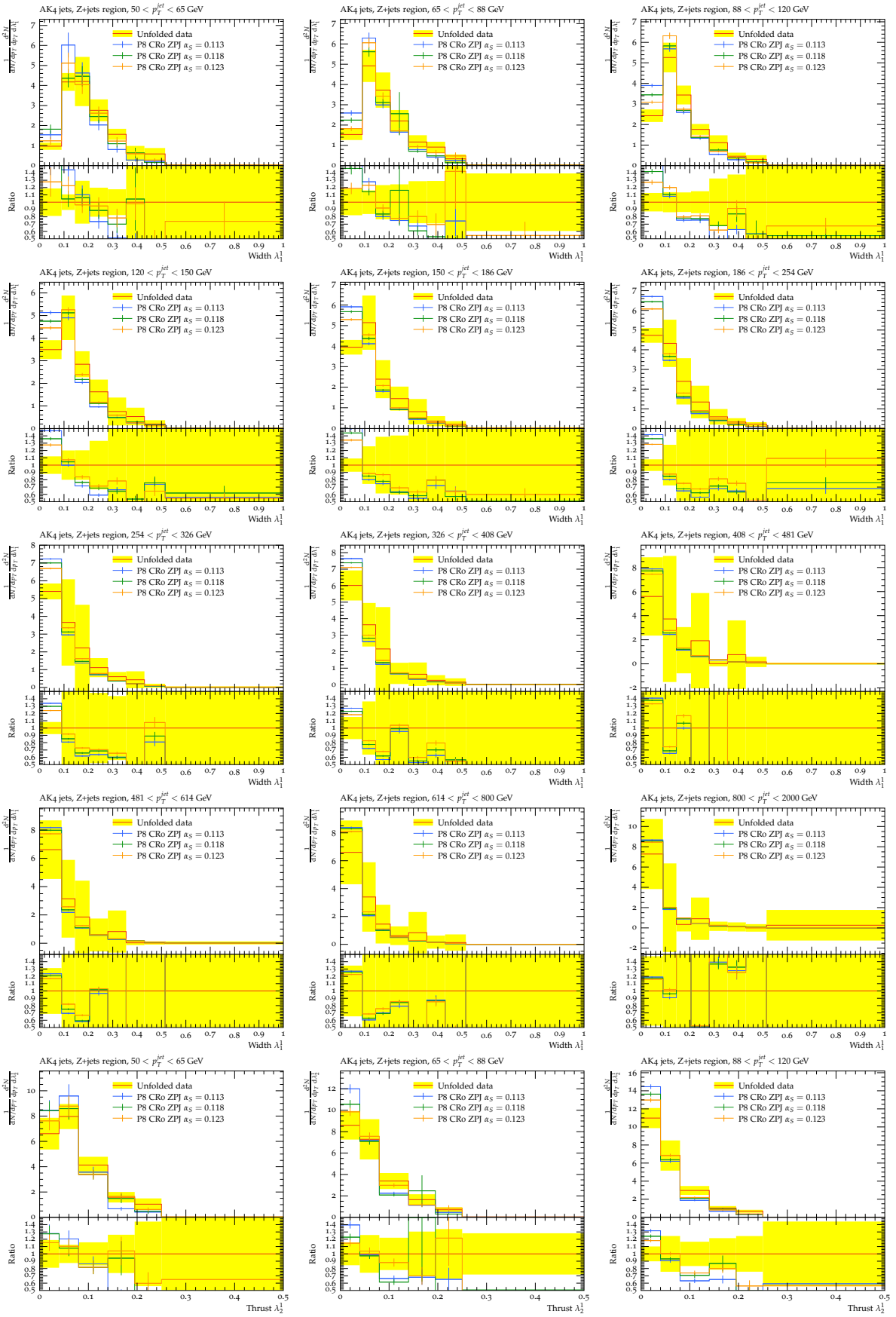


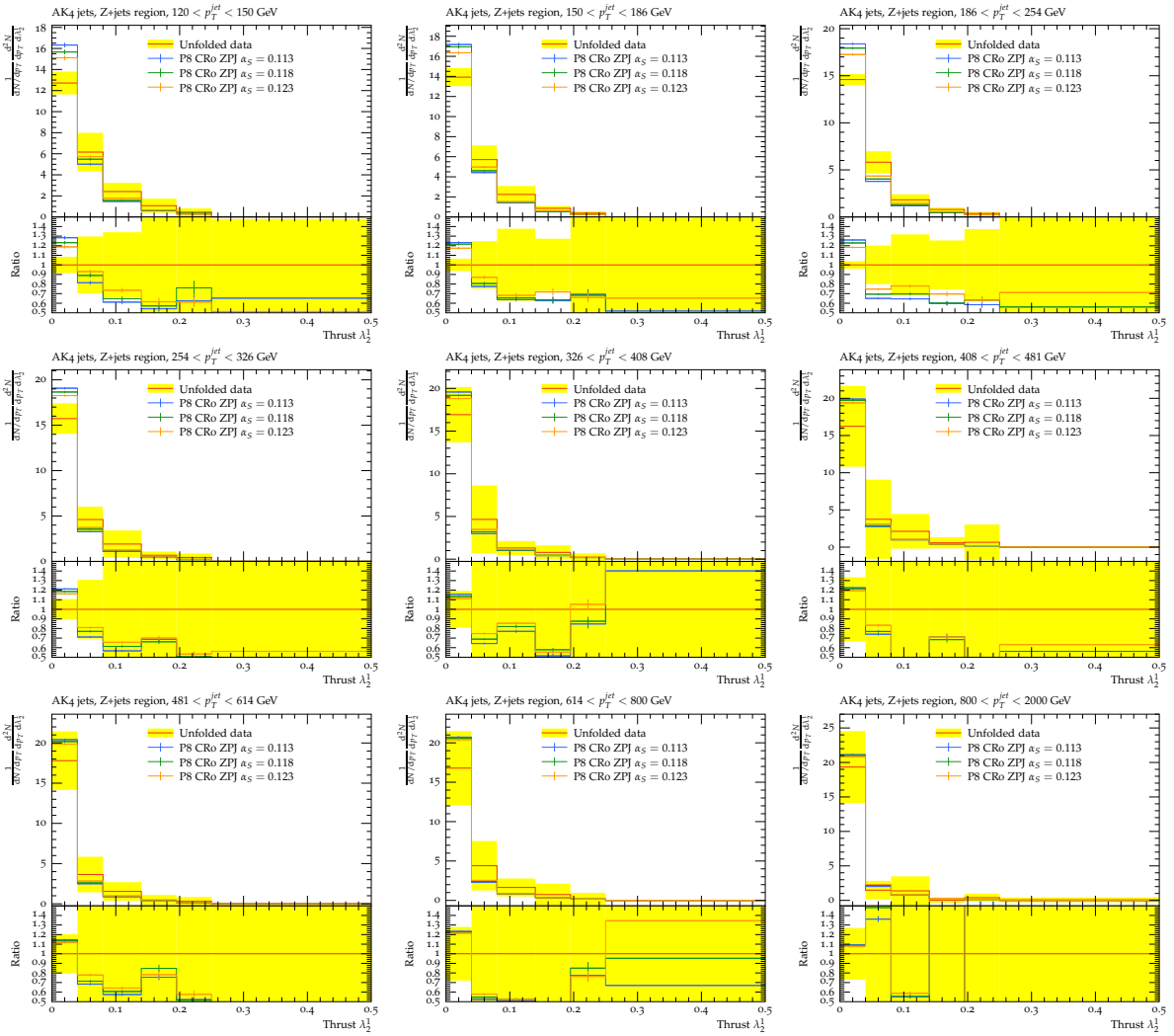
A.5. Pythia8 CR0 Z+Jet Plots



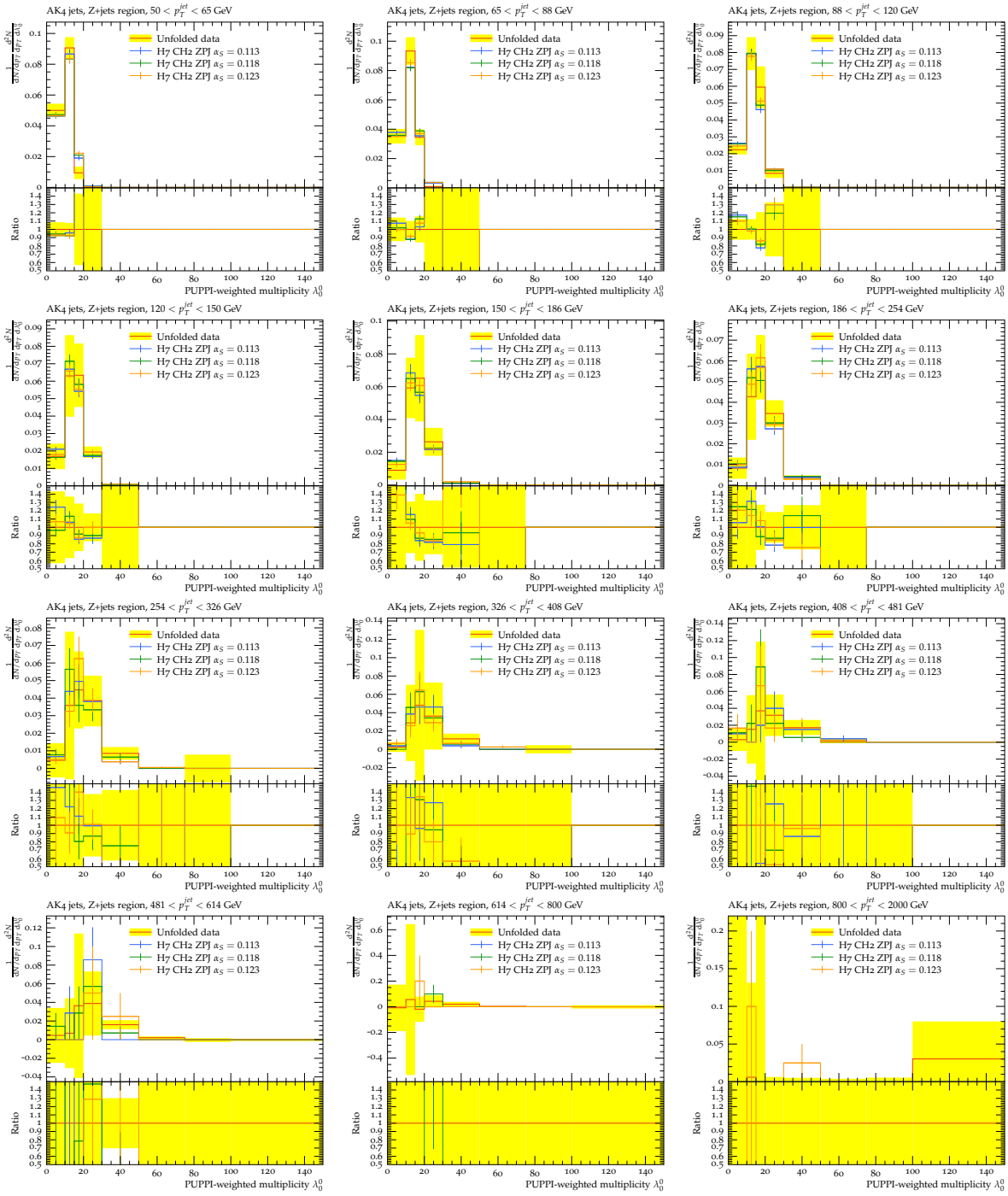

AK4 jets, Z+jets region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Z+jets region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Z+jets region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Z+jets region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Z+jets region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Z+jets region, 800 < p_T^{jet} < 2000 GeV
AK4 jets, Z+jets region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Z+jets region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Z+jets region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Z+jets region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Z+jets region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Z+jets region, 186 < p_T^{jet} < 254 GeV
Unfolded data
P8 CRo ZPJ $\alpha_S = 0.113$
P8 CRo ZPJ $\alpha_S = 0.118$
P8 CRo ZPJ $\alpha_S = 0.123$
Ratio
PUPPI-weighted multiplicity λ_0^0
$(p_T^D)^2$ λ_0^2

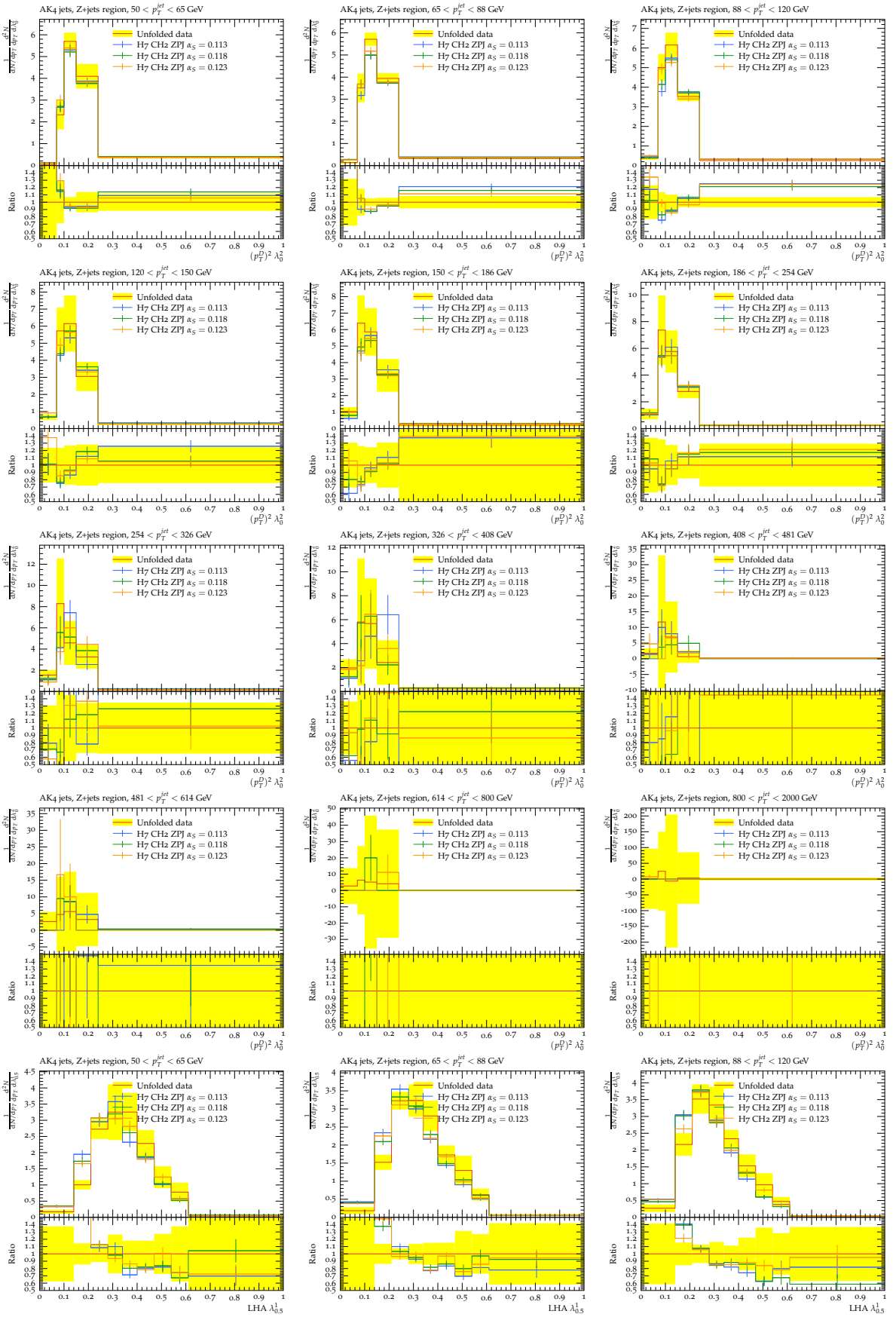

AK4 jets, Z+jets region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Z+jets region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Z+jets region, 800 < p_T^{jet} < 2000 GeV
AK4 jets, Z+jets region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Z+jets region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Z+jets region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Z+jets region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Z+jets region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Z+jets region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Z+jets region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Z+jets region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Z+jets region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Z+jets region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Z+jets region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Z+jets region, 800 < p_T^{jet} < 2000 GeV
Unfolded data
P8 CRo ZPJ $\alpha_S = 0.113$
P8 CRo ZPJ $\alpha_S = 0.118$
P8 CRo ZPJ $\alpha_S = 0.123$
Ratio
$(p_T^D)^2\ \lambda_0^2$
LHA $\lambda_{0.5}^1$

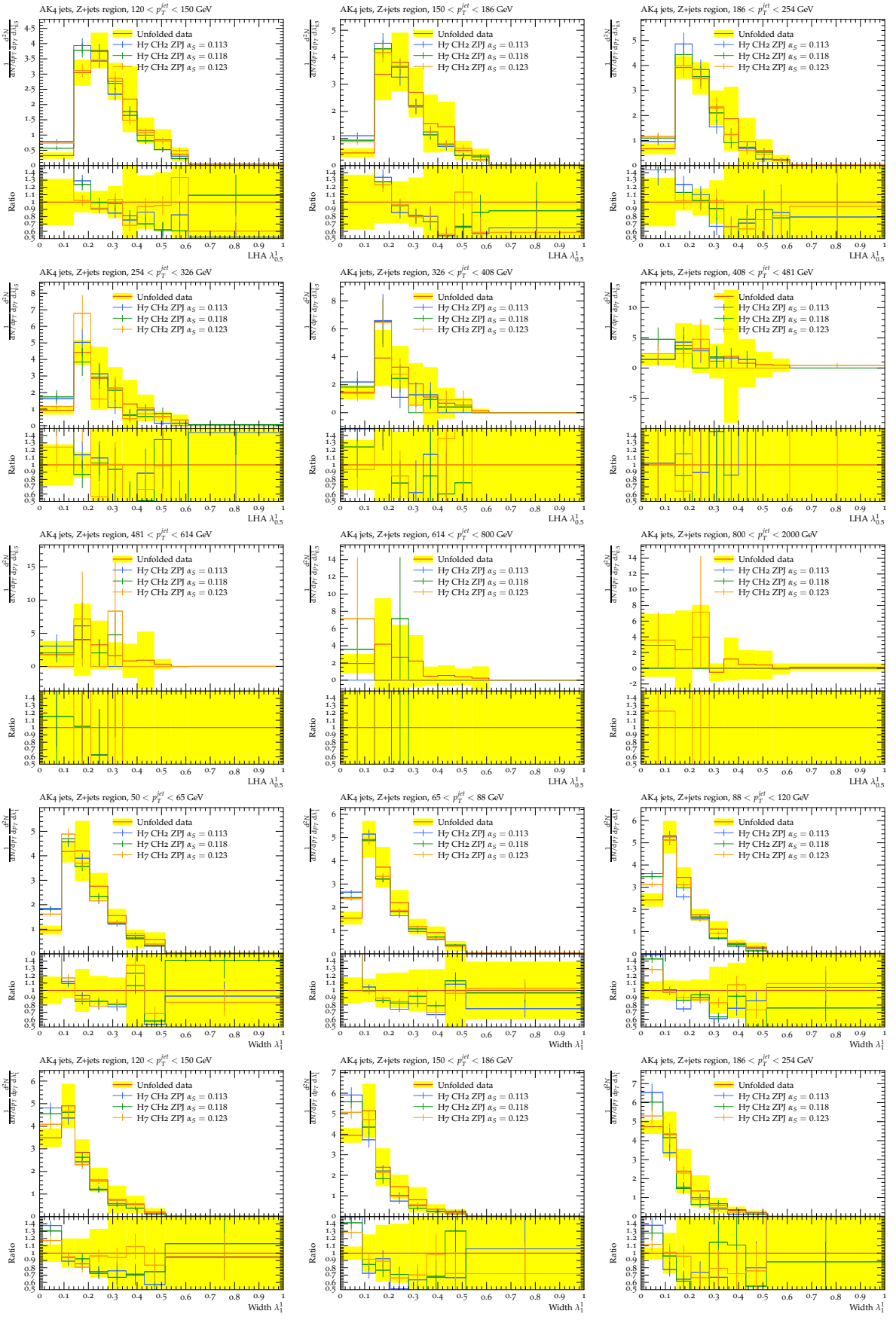

AK4 jets, Z+jets region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Z+jets region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Z+jets region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Z+jets region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Z+jets region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Z+jets region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Z+jets region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Z+jets region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Z+jets region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Z+jets region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Z+jets region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Z+jets region, 800 < p_T^{jet} < 2000 GeV
AK4 jets, Z+jets region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Z+jets region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Z+jets region, 88 < p_T^{jet} < 120 GeV
Unfolded data
P8 CRo ZPJ $\alpha_S = 0.113$
P8 CRo ZPJ $\alpha_S = 0.118$
P8 CRo ZPJ $\alpha_S = 0.123$
Ratio
Width λ_1^1
Thrust λ_2^1

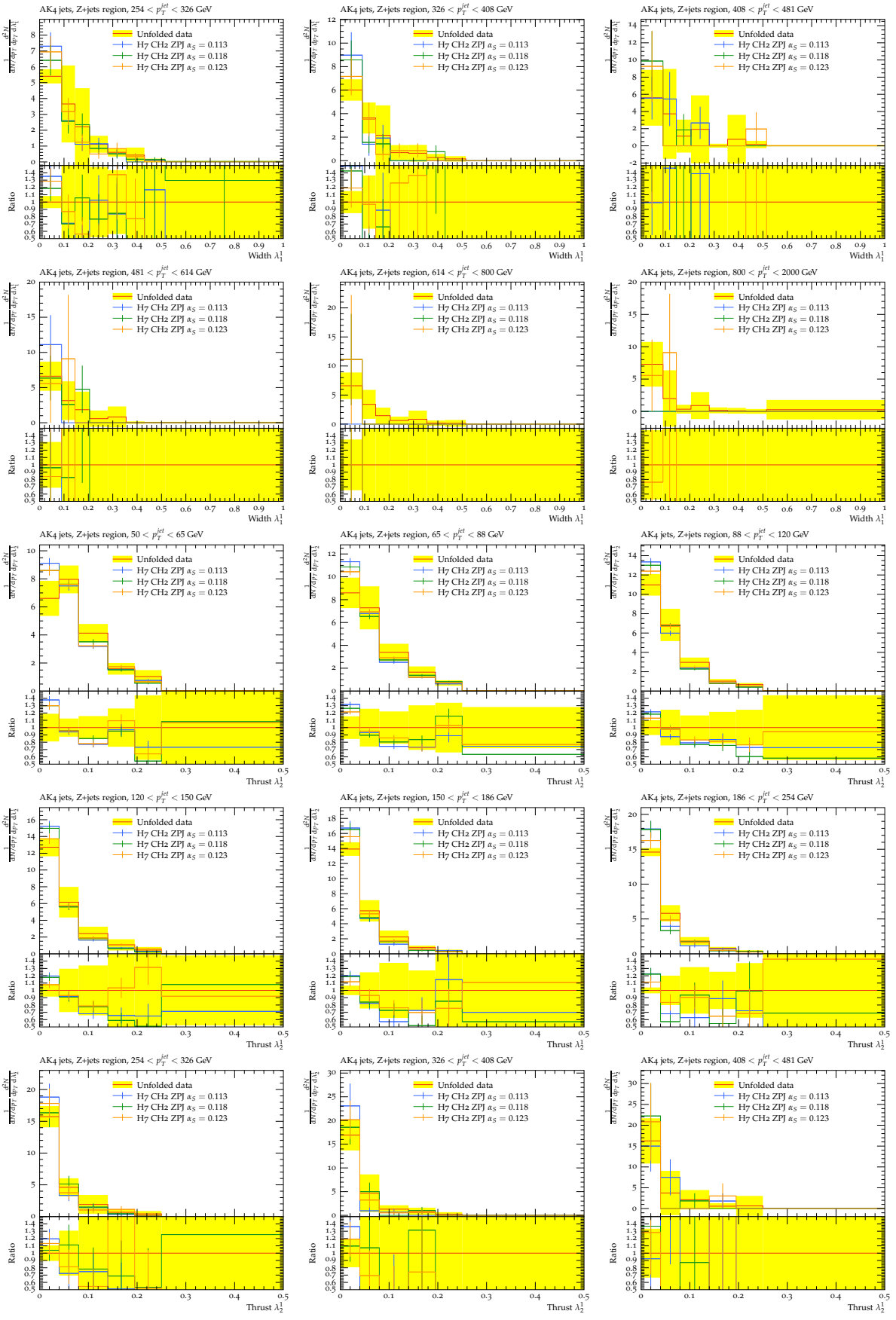

AK4 jets, Z+jets region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Z+jets region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Z+jets region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Z+jets region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Z+jets region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Z+jets region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Z+jets region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Z+jets region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Z+jets region, 800 < p_T^{jet} < 2000 GeV
Unfolded data
P8 CRo ZPJ $\alpha_S = 0.113$
P8 CRo ZPJ $\alpha_S = 0.118$
P8 CRo ZPJ $\alpha_S = 0.123$
Ratio
Thrust λ_2^1

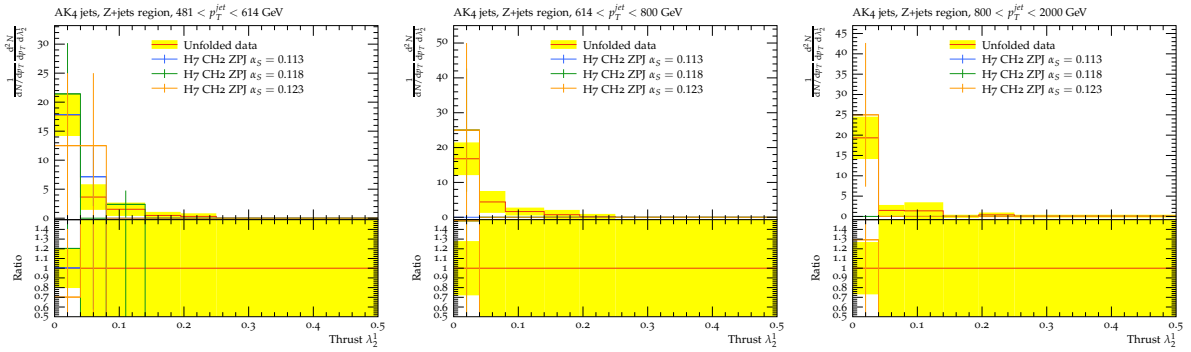
A.6. Herwig7 Z+Jet Plots



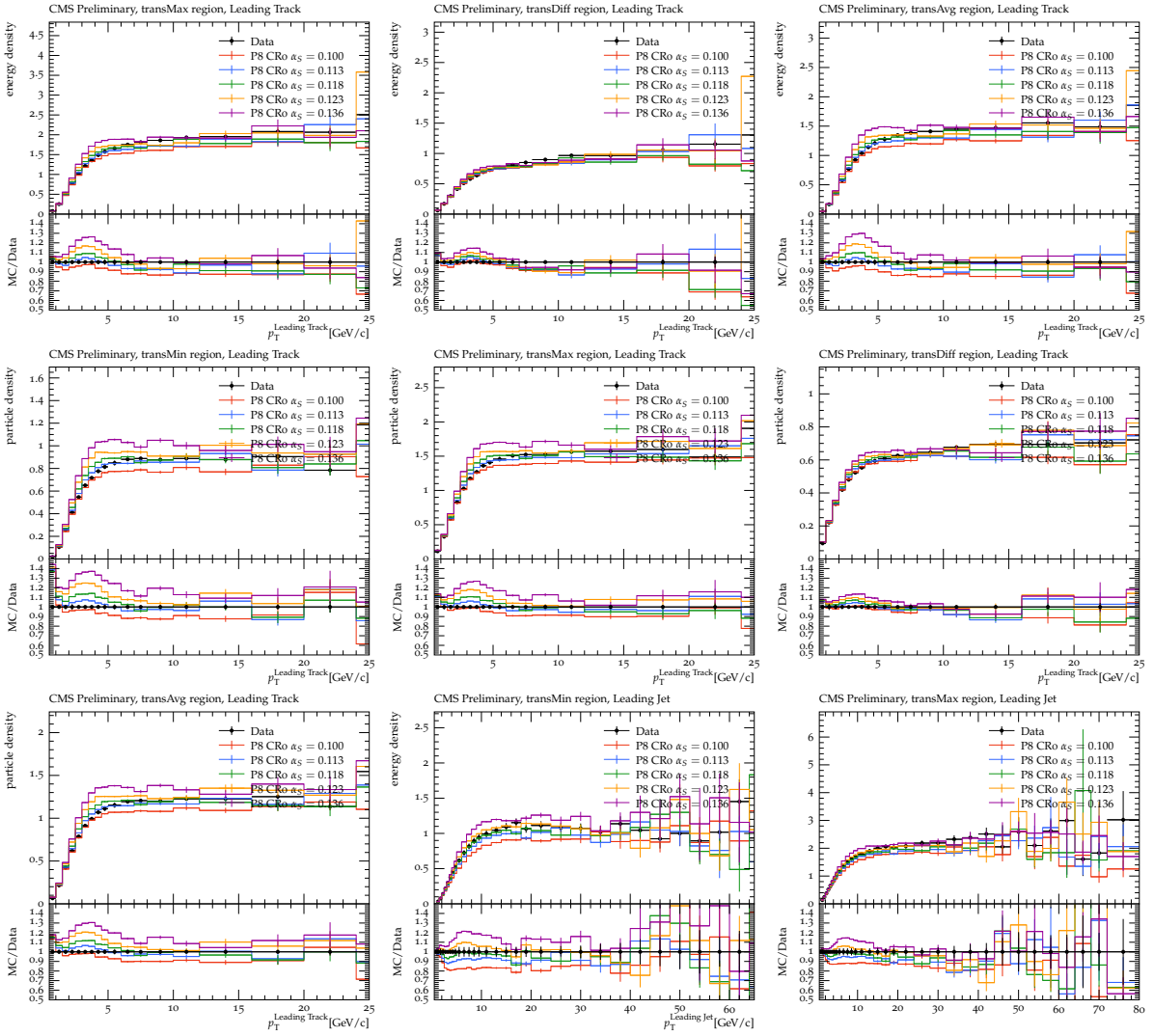

AK4 jets, Z+jets region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Z+jets region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Z+jets region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Z+jets region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Z+jets region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Z+jets region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Z+jets region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Z+jets region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Z+jets region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Z+jets region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Z+jets region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Z+jets region, 800 < p_T^{jet} < 2000 GeV
AK4 jets, Z+jets region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Z+jets region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Z+jets region, 88 < p_T^{jet} < 120 GeV
Unfolded data
H7 CH2 ZPJ $\alpha_S = 0.113$
H7 CH2 ZPJ $\alpha_S = 0.118$
H7 CH2 ZPJ $\alpha_S = 0.123$
Ratio
$(p_T^D)^2\ \lambda_0^2$
LHA $\lambda_{0.5}^1$

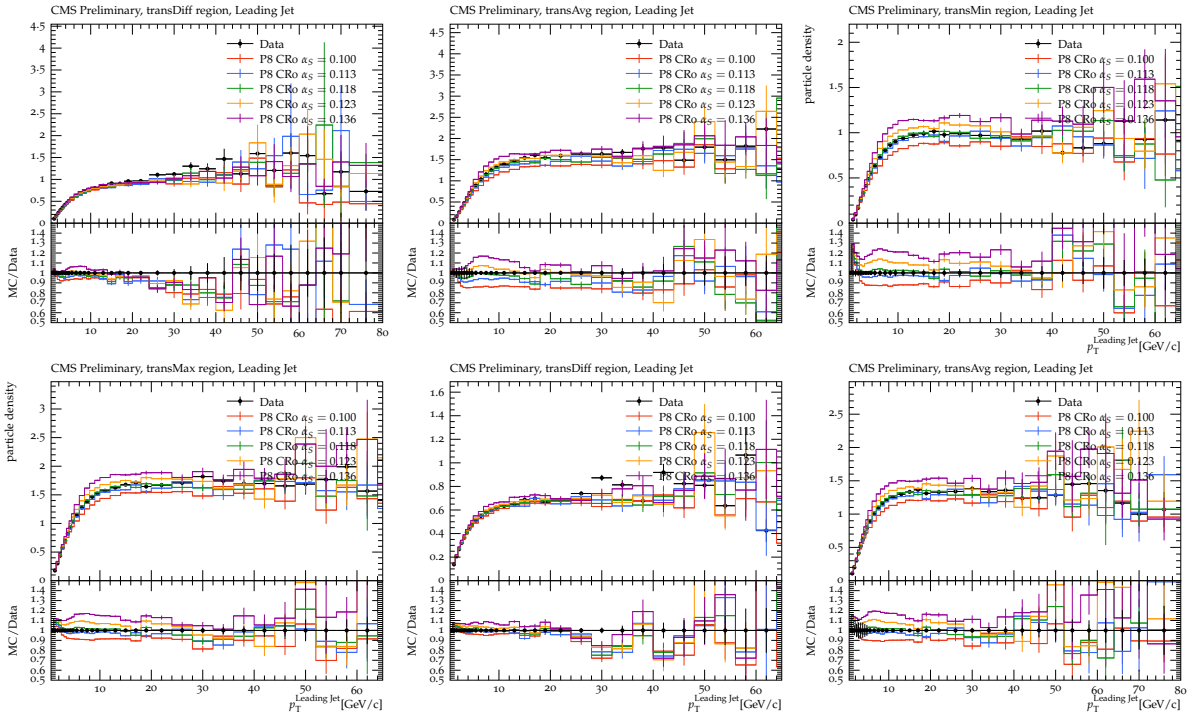

AK4 jets, Z+jets region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Z+jets region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Z+jets region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Z+jets region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Z+jets region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Z+jets region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Z+jets region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Z+jets region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Z+jets region, 800 < p_T^{jet} < 2000 GeV
AK4 jets, Z+jets region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Z+jets region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Z+jets region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Z+jets region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Z+jets region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Z+jets region, 186 < p_T^{jet} < 254 GeV
Unfolded data
H7 CH2 ZPJ $\alpha_S = 0.113$
H7 CH2 ZPJ $\alpha_S = 0.118$
H7 CH2 ZPJ $\alpha_S = 0.123$
Ratio
LHA $\lambda^1_{0.5}$
Width λ^1_1


AK4 jets, Z+jets region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Z+jets region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Z+jets region, 408 < p_T^{jet} < 481 GeV
AK4 jets, Z+jets region, 481 < p_T^{jet} < 614 GeV
AK4 jets, Z+jets region, 614 < p_T^{jet} < 800 GeV
AK4 jets, Z+jets region, 800 < p_T^{jet} < 2000 GeV
AK4 jets, Z+jets region, 50 < p_T^{jet} < 65 GeV
AK4 jets, Z+jets region, 65 < p_T^{jet} < 88 GeV
AK4 jets, Z+jets region, 88 < p_T^{jet} < 120 GeV
AK4 jets, Z+jets region, 120 < p_T^{jet} < 150 GeV
AK4 jets, Z+jets region, 150 < p_T^{jet} < 186 GeV
AK4 jets, Z+jets region, 186 < p_T^{jet} < 254 GeV
AK4 jets, Z+jets region, 254 < p_T^{jet} < 326 GeV
AK4 jets, Z+jets region, 326 < p_T^{jet} < 408 GeV
AK4 jets, Z+jets region, 408 < p_T^{jet} < 481 GeV
Unfolded data
H7 CH2 ZPJ $\alpha_S = 0.113$
H7 CH2 ZPJ $\alpha_S = 0.118$
H7 CH2 ZPJ $\alpha_S = 0.123$
Ratio
Width λ_1^1
Thrust λ_2^1

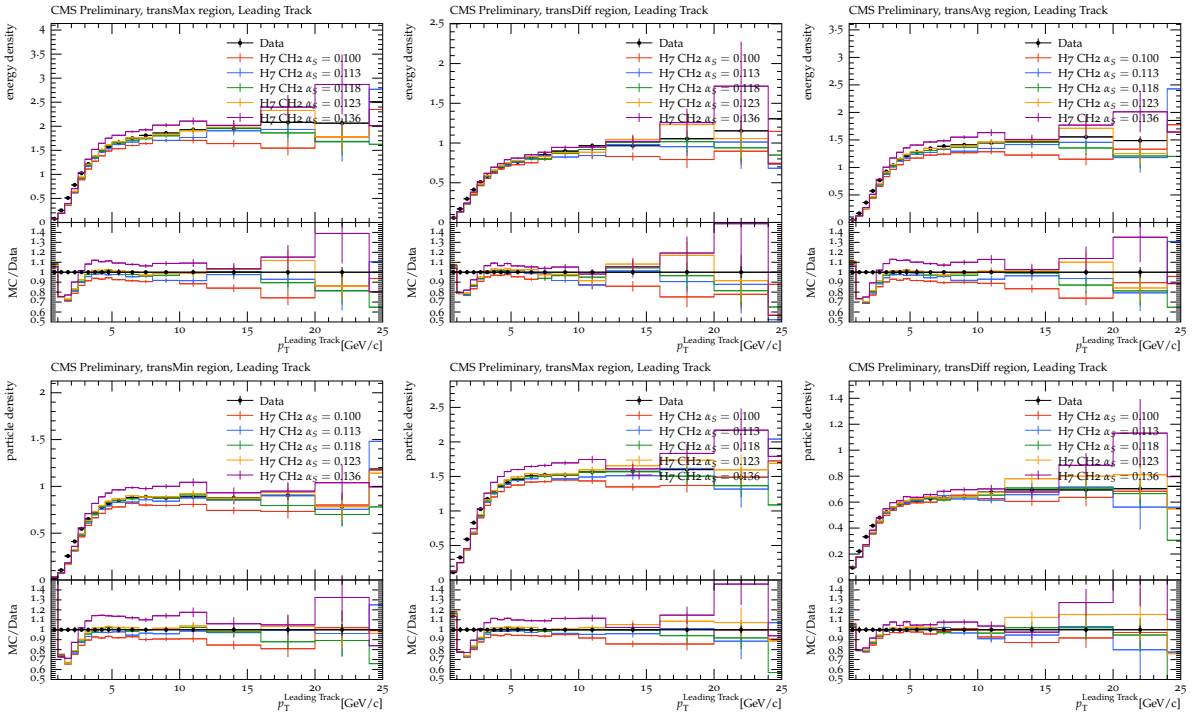


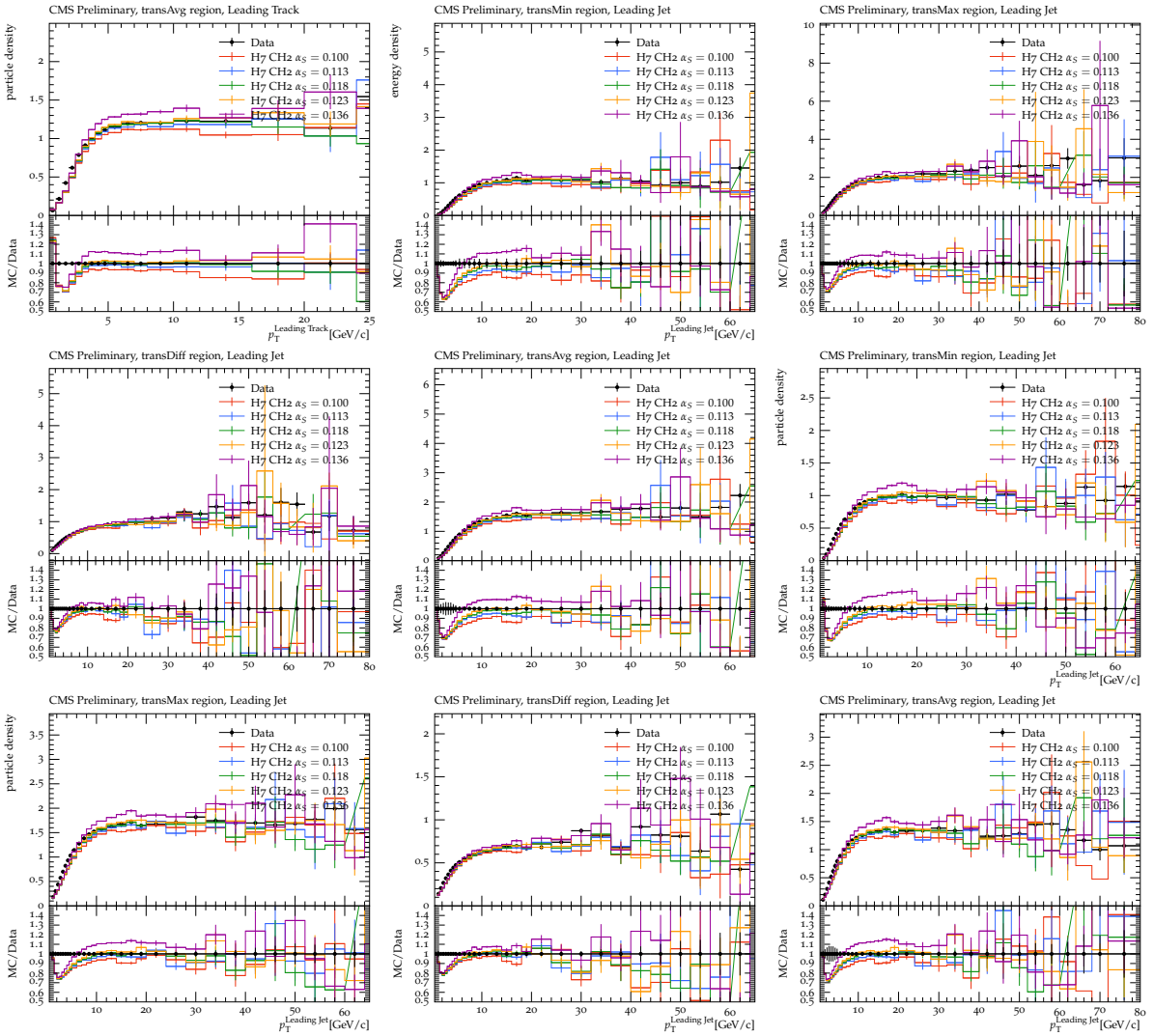
A.7. Pythia8 CR0 UE Plots





A.8. Herwig7 UE Plots




CMS Preliminary, transAvg region, Leading Track
CMS Preliminary, transMin region, Leading Jet
CMS Preliminary, transMax region, Leading Jet
CMS Preliminary, transDiff region, Leading Jet
CMS Preliminary, transAvg region, Leading Jet
CMS Preliminary, transMin region, Leading Jet
CMS Preliminary, transMax region, Leading Jet
CMS Preliminary, transDiff region, Leading Jet
CMS Preliminary, transAvg region, Leading Jet
Data
H7 CH2 $\alpha_S = 0.100$
H7 CH2 $\alpha_S = 0.113$
H7 CH2 $\alpha_S = 0.118$
H7 CH2 $\alpha_S = 0.123$
H7 CH2 $\alpha_S = 0.136$
particle density
energy density
MC/Data
$p_T^{\text{Leading Track}}$ [GeV/c]
$p_T^{\text{Leading Jet}}$ [GeV/c]

Appendix B.

Analysis Files

B.1. Dijet Analysis

```
// -*- C++ -*-
#include "Rivet/Analysis.hh"
#include "Rivet/Projections/FinalState.hh"
#include "Rivet/Projections/FastJets.hh"
#include "Rivet/Projections/IdentifiedFinalState.hh"
#include "Rivet/Projections/VetoedFinalState.hh"
#include "Rivet/Jet.hh"

#include "fastjet/JetDefinition.hh"
#include "fastjet/ClusterSequence.hh"
#include "fastjet/tools/Recluster.hh"
#include "fastjet/contrib/SoftDrop.hh"
//#include "fastjet/contrib/ModifiedMassDropTagger.hh"

#include <algorithm>


using std::cout;
using std::endl;
using std::vector;
using namespace fastjet;


namespace Rivet {

/// \class Angularity
/// definition of angularity
///
class Angularity : public FunctionOfPseudoJet<double>{
public:
  /// ctor
  Angularity(double alpha, double jet_radius, double kappa=1.0, bool isCharged=false, Selector constitCut=SelectorPtMin(0.))
        : _alpha(alpha), _radius(jet_radius), _kappa(kappa), _isCharged(isCharged), _constitCut(constitCut) {}

  /// description
  std::string description() const{
    ostringstream oss;
    oss << "Angularity with alpha=" << _alpha;
    return oss.str();
  }

  /// computation of the angularity itself
  double result(const PseudoJet &jet) const{
    // get the jet constituents
    vector<PseudoJet> constits = jet.constituents();
```

```
    // get the reference axis
    PseudoJet reference_axis = _get_reference_axis(jet);

    // do the actual coputation
    double numerator = 0.0, denominator = 0.0;
    unsigned int num = 0;
    for (const auto &c : constits){
      if ((_isCharged) && (!c.user_index())) continue;
      if (!_constitCut.pass(c)) continue;
      double pt = c.pt();
      // Note: better compute (dist^2)^(alpha/2) to avoid an extra square root
      numerator   += pow(pt, _kappa) * pow(c.squared_distance(reference_axis), 0.5*_alpha);
      denominator += pt;
      num += 1;
    }
    if (!((num >= _minNumConstits) && (denominator > 0))) return -1;
    // the formula is only correct for the the typical angularities which satisfy either kappa==1 or beta==0.
    else return numerator/(pow(denominator, _kappa)*pow(_radius, _alpha));
  }

protected:
  PseudoJet _get_reference_axis(const PseudoJet &jet) const{
    if (_alpha>1) return jet;

    Recluster recluster(JetDefinition(antikt_algorithm, JetDefinition::max_allowable_R, WTA_pt_scheme));
    return recluster(jet);
  }

  double _alpha, _radius, _kappa;
  bool _isCharged;
  Selector _constitCut;
  const uint _minNumConstits = 2;
};

/**
 * Lightweight class to hold info about Lambda variable
 */
  class LambdaVar {

  public:
    LambdaVar(const std::string & name_, float kappa_, float beta_, bool isCharged_, Selector constitCut_):
      name(name_),
      kappa(kappa_),
      beta(beta_),
      isCharged(isCharged_),
      constitCut(constitCut_)
    {}

    std::string name;
    float kappa;
    float beta;
    bool isCharged;
    Selector constitCut;
  };


  /// @brief Routine for QG substructure analysis
  class CMS_2018_PAS_SMP_18_QGX_DIJET : public Analysis {
  public:

    /// Constructor
    DEFAULT_RIVET_ANALYSIS_CTOR(CMS_2018_PAS_SMP_18_QGX_DIJET);

    /// Book histograms and initialise projections before the run
    void init() {

      // Initialise and register projections
      // Particles for the jets
      FinalState fs(-5, 5, 0.0*GeV);
      VetoedFinalState jet_input(fs);
      jet_input.vetoNeutrinos();
      addProjection(jet_input, "JET_INPUT");

      // Book histograms
```

```
  // resize vectors appropriately
  uint nHistsRadii = _jetRadii.size();
  uint nHistsLambda = _lambdaVars.size();
  uint nHistsPt = _ptBinsGen.size()-1;
  _h_dijet_cen.resize(nHistsRadii, vector<vector<Histo1DPtr> >(nHistsLambda, vector<Histo1DPtr>(nHistsPt)));
  _h_dijet_cen_groomed.resize(nHistsRadii, vector<vector<Histo1DPtr> >(nHistsLambda, vector<Histo1DPtr>(nHistsPt)));
  _h_dijet_fwd.resize(nHistsRadii, vector<vector<Histo1DPtr> >(nHistsLambda, vector<Histo1DPtr>(nHistsPt)));
  _h_dijet_fwd_groomed.resize(nHistsRadii, vector<vector<Histo1DPtr> >(nHistsLambda, vector<Histo1DPtr>(nHistsPt)));

  // Now book histos
  // remember 1-indexed

  // yoda plot naming scheme
  // ----------------------------------------------------------------------
  // d = channel (ak4/8 radii (10, 20) + {dijet cen / fwd} * groomed versions[1..4])
  // x = lambda variable; neutral+charged & charged-only are treated separately
  // y = pT bin
  for (uint radiusInd=1; radiusInd <= _jetRadii.size(); radiusInd++) {
    for (uint lambdaInd=1; lambdaInd <= _lambdaVars.size(); lambdaInd++) {
      for (uint ptInd=1; ptInd < _ptBinsGen.size(); ptInd++) {
        _h_dijet_cen[radiusInd-1][lambdaInd-1][ptInd-1] = bookHisto1D((10*radiusInd) + 1, lambdaInd, ptInd);
        _h_dijet_cen_groomed[radiusInd-1][lambdaInd-1][ptInd-1] = bookHisto1D((10*radiusInd) + 2, lambdaInd, ptInd);
        _h_dijet_fwd[radiusInd-1][lambdaInd-1][ptInd-1] = bookHisto1D((10*radiusInd) + 3, lambdaInd, ptInd);
        _h_dijet_fwd_groomed[radiusInd-1][lambdaInd-1][ptInd-1] = bookHisto1D((10*radiusInd) + 4, lambdaInd, ptInd);
      }
    }
  }

  // _h_pt_jet = bookHisto1D(1, 90, 1);

  vector<double> binEdges = {
    1E-9,
    5E-9,
    1E-8,
    5E-8,
    1E-7,
    5E-7,
    1E-6,
    5E-6,
    1E-5,
    5E-5,
    1E-4,
    5E-4,
    1E-3,
    5E-3,
    1E-2,
    5E-2,
    1E-1,
    5E-1,
    1,
    5,
    10,
  };

}


// Get index of largest bin smaller than value in vector
// e.g. what you'd need when binning a continuous variable
uint getBinIndex(float value, const vector<float> & bins) {
  auto itr = std::lower_bound(bins.begin(), bins.end(), value);
  return itr - bins.begin() - 1;
}


/// Perform the per-event analysis
void analyze(const Event& event) {
  const double weight = event.weight();

  // Convert Particles into PseudoJets for clustering
  const VetoedFinalState & fs = applyProjection<VetoedFinalState>(event, "JET_INPUT");
  const ParticleVector & fsParticles = fs.particles();
  vector<PseudoJet> particles;
  particles.reserve(fsParticles.size());
```

```
for (uint iFS=0; iFS<fsParticles.size(); iFS++){
  PseudoJet p = fsParticles[iFS].pseudojet();
  p.set_user_index(fsParticles[iFS].isCharged()); // for later reference to charge
  particles.push_back(p);
}

for (uint radiusInd=0; radiusInd < _jetRadii.size(); radiusInd++) {
  float jetRadius = _jetRadii.at(radiusInd);

  JetDefinition jet_def(antikt_algorithm, jetRadius);
  vector<PseudoJet> jets = (SelectorNHardest(2) * SelectorAbsRapMax(1.7) * SelectorPtMin(15))(jet_def(particles));

  bool passDijet = false;
  if (jets.size() < 2) continue;
  const auto & jet1 = jets.at(0);
  const auto & jet2 = jets.at(1);
  float jet1pt = jet1.pt();
  float jet2pt = jet2.pt();
  float asym = (jet1pt - jet2pt) / (jet1pt+jet2pt);
  float dphi = Rivet::deltaPhi(jet1.phi(), jet2.phi());
  passDijet = ((asym < 0.3) && (dphi > 2.0));

  if (!passDijet) continue;

  // Sort by increasing absolute rapidity
  vector<PseudoJet> dijets = {jet1, jet2};
  std::sort(dijets.begin(), dijets.end(),
            [] (const PseudoJet & A, const PseudoJet & B)
            { return fabs(A.rapidity()) < fabs(B.rapidity()); }
           );

  for (uint iJ=0; iJ<dijets.size(); iJ++) {
    bool isCentral = (iJ == 0);
    PseudoJet & jetItr = dijets[iJ];

    // _h_pt_jet->fill(jetItr.pt(), weight);

    // Simplify life - ignore this jet if it is below 1st hist pt range
    // Note that we don't apply it to the original jet pt cut - since
    // we have phase space where one jet is > 50, and one < 50
    if (jetItr.pt() < _ptBinsGen[0]) continue;
    // ignore jet if beyond the last bin
    if (jetItr.pt() > _ptBinsGen.back()) continue;

    // Need to use original, ungroomed jet pT to bin
    uint ptBinInd = getBinIndex(jetItr.pt(), _ptBinsGen);

    // UNGROOMED VERSION
    // ----------------------------------------------------------------------

    // Fill hists for each lambda variable
    for (uint lambdaInd=0; lambdaInd < _lambdaVars.size(); lambdaInd++) {
      const LambdaVar & thisLambdaVar = _lambdaVars[lambdaInd];
      Angularity angularity(thisLambdaVar.beta, jetRadius, thisLambdaVar.kappa, thisLambdaVar.isCharged, thisLambdaVar
          .constitCut);
      float val = angularity(jetItr);
      if (val<0) continue;

      if (isCentral) {
        _h_dijet_cen[radiusInd][lambdaInd][ptBinInd]->fill(val, weight);
      } else {
        _h_dijet_fwd[radiusInd][lambdaInd][ptBinInd]->fill(val, weight);
      }
    }

    // GROOMED VERSION
    // ----------------------------------------------------------------------
    // Get groomed jet
    fastjet::contrib::SoftDrop sd(0, 0.1, jetRadius);
    PseudoJet groomedJet = sd(jetItr);
    //fastjet::contrib::ModifiedMassDropTagger mmdt(0.1); mmdt.set_grooming_mode(); mmdt.set_reclustering(false);
    //Recluster ca_cluster(JetDefinition(cambridge_algorithm, JetDefinition::max_allowable_R));
    //PseudoJet groomedJet = mmdt(ca_cluster(jetItr));
```

```
        // Fill hists for each lambda variable
        for (uint lambdaInd=0; lambdaInd < _lambdaVars.size(); lambdaInd++) {
          const LambdaVar & thisLambdaVar = _lambdaVars[lambdaInd];
          Angularity angularity(thisLambdaVar.beta, jetRadius, thisLambdaVar.kappa, thisLambdaVar.isCharged, thisLambdaVar
              .constitCut);
          float val = angularity(groomedJet);
          if (val<0) continue;

          if (isCentral) {
            _h_dijet_cen_groomed[radiusInd][lambdaInd][ptBinInd]->fill(val, weight);
          } else {
            _h_dijet_fwd_groomed[radiusInd][lambdaInd][ptBinInd]->fill(val, weight);
          }
        }

      } // end loop over dijets
    } // end loop over jet radii
  } // end analyze() function


  /// Normalise histograms etc., after the run
  void finalize() {

    // norm hists to cross section
    // scale(_h_pt_jet, lumi * crossSection() / sumOfWeights());

  } // end of finalize

  // Order matters here
  const vector<float> _jetRadii = {0.4, 0.8};

  // This order is important! index in vector used to create YODA plot name
  // Must match that in extracRivetPlotsDijet.py
  const vector<LambdaVar> _lambdaVars = {
    LambdaVar("jet_multiplicity", 0, 0, false, SelectorPtMin(1.)),
    LambdaVar("jet_pTD", 2, 0, false, SelectorPtMin(0.)),
    LambdaVar("jet_LHA", 1, 0.5, false, SelectorPtMin(0.)),
    LambdaVar("jet_width", 1, 1, false, SelectorPtMin(0.)),
    LambdaVar("jet_thrust", 1, 2, false, SelectorPtMin(0.)),
    LambdaVar("jet_multiplicity_charged", 0, 0, true, SelectorPtMin(1.)),
    LambdaVar("jet_pTD_charged", 2, 0, true, SelectorPtMin(0.)),
    LambdaVar("jet_LHA_charged", 1, 0.5, true, SelectorPtMin(0.)),
    LambdaVar("jet_width_charged", 1, 1, true, SelectorPtMin(0.)),
    LambdaVar("jet_thrust_charged", 1, 2, true, SelectorPtMin(0.)),
  };

  const vector<float> _ptBinsGen = {
    50, 65, 88, 120, 150, 186, 254, 326, 408, 481, 614, 800, 1000, 4000
  };

  /// @name Histograms
  Histo1DPtr _h_pt_jet;

  // 3D vector: [jet radius][lambda variable][pt bin]
  // since each pt bin has its own normalised distribution
  vector<vector<vector<Histo1DPtr> > > _h_dijet_cen,
                                       _h_dijet_cen_groomed,
                                       _h_dijet_fwd,
                                       _h_dijet_fwd_groomed;

};


// The hook for the plugin system
DECLARE_RIVET_PLUGIN(CMS_2018_PAS_SMP_18_QGX_DIJET);


}
```

B.2. Z+Jet Analysis

```
// -*- C++ -*-
#include "Rivet/Analysis.hh"
#include "Rivet/Projections/FinalState.hh"
#include "Rivet/Projections/FastJets.hh"
#include "Rivet/Projections/IdentifiedFinalState.hh"
#include "Rivet/Projections/VetoedFinalState.hh"
#include "Rivet/Projections/ZFinder.hh"
#include "Rivet/Projections/IdentifiedFinalState.hh"
#include "Rivet/Jet.hh"

#include "fastjet/JetDefinition.hh"
#include "fastjet/ClusterSequence.hh"
#include "fastjet/tools/Recluster.hh"
#include "fastjet/contrib/SoftDrop.hh"
//#include "fastjet/contrib/ModifiedMassDropTagger.hh"

#include <algorithm>


using std::cout;
using std::endl;
using std::vector;
using namespace fastjet;


namespace Rivet {

/// \class Angularity
/// definition of angularity
///
class Angularity : public FunctionOfPseudoJet<double>{
public:
  /// ctor
  Angularity(double alpha, double jet_radius, double kappa=1.0, bool isCharged=false, Selector constitCut=SelectorPtMin(0.))
        : _alpha(alpha), _radius(jet_radius), _kappa(kappa), _isCharged(isCharged), _constitCut(constitCut) {}

  /// description
  std::string description() const{
    ostringstream oss;
    oss << "Angularity with alpha=" << _alpha;
    return oss.str();
  }

  /// computation of the angularity itself
  double result(const PseudoJet &jet) const{
    // get the jet constituents
    vector<PseudoJet> constits = jet.constituents();

    // get the reference axis
    PseudoJet reference_axis = _get_reference_axis(jet);

    // do the actual coputation
    double numerator = 0.0, denominator = 0.0;
    unsigned int num = 0;
    for (const auto &c : constits){
      if ((_isCharged) && (!c.user_index())) continue;
      if (!_constitCut.pass(c)) continue;
      double pt = c.pt();
      // Note: better compute (dist^2)^(alpha/2) to avoid an extra square root
      numerator   += pow(pt, _kappa) * pow(c.squared_distance(reference_axis), 0.5*_alpha);
      denominator += pt;
      num += 1;
    }
    if (!((num >= _minNumConstits) && (denominator > 0))) return -1;
    // the formula is only correct for the the typical angularities which satisfy either kappa==1 or beta==0.
    else return numerator/(pow(denominator, _kappa)*pow(_radius, _alpha));
  }

protected:
  PseudoJet _get_reference_axis(const PseudoJet &jet) const{
    if (_alpha>1) return jet;
```

```
    Recluster recluster(JetDefinition(antikt_algorithm, JetDefinition::max_allowable_R, WTA_pt_scheme));
    return recluster(jet);
  }

  double _alpha, _radius, _kappa;
  bool _isCharged;
  Selector _constitCut;
  const uint _minNumConstits = 2;
};


/**
 * Lightweight class to hold info about Lambda variable
 */
  class LambdaVar {

  public:
    LambdaVar(const std::string & name_, float kappa_, float beta_, bool isCharged_, Selector constitCut_):
      name(name_),
      kappa(kappa_),
      beta(beta_),
      isCharged(isCharged_),
      constitCut(constitCut_)
    {}

    std::string name;
    float kappa;
    float beta;
    bool isCharged;
    Selector constitCut;
  };


  /// @brief Routine for QG substructure analysis
  class CMS_2018_PAS_SMP_18_QGX_ZPJ : public Analysis {
  public:

    /// Constructor
    DEFAULT_RIVET_ANALYSIS_CTOR(CMS_2018_PAS_SMP_18_QGX_ZPJ);

    /// Book histograms and initialise projections before the run
    void init() {

      // Initialise and register projections
      // Particles for the jets
      FinalState fs(-5, 5, 0.0*GeV);
      VetoedFinalState jet_input(fs);
      jet_input.vetoNeutrinos();
      addProjection(jet_input, "JET_INPUT");

      // for the muons
      double mu_pt = 26.;
      double mz_min = (90-20);
      double mz_max = (90+20);
      double eta_max = 2.4;
      ZFinder zfinder(fs,
                      Cuts::pT > mu_pt*GeV  && Cuts::abseta < eta_max,
                      PID::MUON,
                      mz_min*GeV, mz_max*GeV,
                      0.1, ZFinder::NOCLUSTER, ZFinder::NOTRACK);
      addProjection(zfinder, "ZFinder");

      eta_max = 2.4;
      FinalState fs_muons(-eta_max, eta_max, 0*GeV);
      IdentifiedFinalState muons_noCut(fs_muons, {PID::MUON, PID::ANTIMUON});
      addProjection(muons_noCut, "MUONS_NOCUT");

      // Book histograms
      // resize vectors appropriately
      uint nHistsRadii = _jetRadii.size();
      uint nHistsLambda = _lambdaVars.size();
      uint nHistsPt = _ptBinsGen.size()-1;
      _h_zpj.resize(nHistsRadii, vector<vector<Histo1DPtr> >(nHistsLambda, vector<Histo1DPtr>(nHistsPt)));
```

```
  _h_zpj_groomed.resize(nHistsRadii, vector<vector<Histo1DPtr> >(nHistsLambda, vector<Histo1DPtr>(nHistsPt)));

  // Now book histos
  // remember 1-indexed

  // yoda plot naming scheme
  // ----------------------------------------------------------------------
  // d = channel (ak4/8 radii (10, 20) + {dijet cen / fwd} * groomed versions[1..4])
  // x = lambda variable; neutral+charged & charged-only are treated separately
  // y = pT bin
  for (uint radiusInd=1; radiusInd <= _jetRadii.size(); radiusInd++) {
    for (uint lambdaInd=1; lambdaInd <= _lambdaVars.size(); lambdaInd++) {
      for (uint ptInd=1; ptInd < _ptBinsGen.size(); ptInd++) {
        _h_zpj[radiusInd-1][lambdaInd-1][ptInd-1] = bookHisto1D((10*radiusInd) + 1, lambdaInd, ptInd);
        _h_zpj_groomed[radiusInd-1][lambdaInd-1][ptInd-1] = bookHisto1D((10*radiusInd) + 2, lambdaInd, ptInd);
      }
    }
  }

  std::vector<double> weightBinEdges = {
    1E-9,
    5E-9,
    1E-8,
    5E-8,
    1E-7,
    5E-7,
    1E-6,
    5E-6,
    1E-5,
    5E-5,
    1E-4,
    5E-4,
    1E-3,
    5E-3,
    1E-2,
    5E-2,
    1E-1,
    5E-1,
    1,
    5,
    10,
  };

  // before selection cuts
  _h_n_muons_no_cut = bookHisto1D("n_muons_no_cut", 3, 0, 3);
  _h_pt_z_no_cut = bookHisto1D("pt_z_no_cut", 200, 0, 1000);
  _h_pt_mu_no_cut = bookHisto1D("pt_mu_no_cut", 200, 0, 1000);
  _h_eta_mu_no_cut = bookHisto1D("eta_mu_no_cut", 50, -5, 5);
  _h_pt_jet_no_cut = bookHisto1D("pt_jet_no_cut", 200, 0, 1000);
  _h_eta_jet_no_cut = bookHisto1D("eta_jet_no_cut", 50, -5, 5);
  _h_dphi_jet_z_no_cut = bookHisto1D("dphi_jet_z_no_cut", 64, 0, 3.2);

  // after selection cuts
  _h_pt_z = bookHisto1D("pt_z", 200, 0, 1000);
  _h_pt_mu = bookHisto1D("pt_mu", 200, 0, 1000);
  _h_eta_mu = bookHisto1D("eta_mu", 50, -5, 5);
  // _h_pt_jet = bookHisto1D("pt_jet", 200, 0, 1000);
  // _h_pt_jet = bookHisto1D(1, 90, 1);
  _h_eta_jet = bookHisto1D("eta_jet", 50, -5, 5);
  _h_dphi_jet_z = bookHisto1D("dphi_jet_z", 64, 0, 3.2);

}


uint getBinIndex(float value, const std::vector<float> & bins) {
  auto itr = std::lower_bound(bins.begin(), bins.end(), value);
  return itr - bins.begin() - 1;
}


/// Perform the per-event analysis
void analyze(const Event& event) {
  const double weight = event.weight();
```

```
// Convert Particles into PseudoJets for clustering
const VetoedFinalState & fs = applyProjection<VetoedFinalState>(event, "JET_INPUT");
const ParticleVector & fsParticles = fs.particles();
vector<PseudoJet> particles;
particles.reserve(fsParticles.size());
for (uint iFS=0; iFS<fsParticles.size(); iFS++){
  PseudoJet p = fsParticles[iFS].pseudojet();
  p.set_user_index(fsParticles[iFS].isCharged()); // for later reference to charge
  particles.push_back(p);
}

for (uint radiusInd=0; radiusInd < _jetRadii.size(); radiusInd++) {
  float jetRadius = _jetRadii.at(radiusInd);

  JetDefinition jet_def(antikt_algorithm, jetRadius);
  vector<PseudoJet> jets = (SelectorNHardest(1) * SelectorAbsRapMax(1.7) * SelectorPtMin(15))(jet_def(particles));

  const FinalState& muons = applyProjection<IdentifiedFinalState>(event, "MUONS_NOCUT");
  _h_n_muons_no_cut->fill(muons.size(), weight);
  if (muons.size() >= 2) {
    Particle muon1 = muons.particlesByPt()[0];
    Particle muon2 = muons.particlesByPt()[1];
    FourMomentum z = muon1.momentum() + muon2.momentum();
    _h_pt_z_no_cut->fill(z.pt(), weight);
    _h_pt_mu_no_cut->fill(muon1.pt(), weight);
    _h_pt_mu_no_cut->fill(muon2.pt(), weight);
    _h_eta_mu_no_cut->fill(muon1.eta(), weight);
    _h_eta_mu_no_cut->fill(muon2.eta(), weight);

    if (jets.size() > 0) {
      PseudoJet jet1 = jets[0];
      _h_pt_jet_no_cut->fill(jet1.pt(), weight);
      _h_eta_jet_no_cut->fill(jet1.eta(), weight);
      _h_dphi_jet_z_no_cut->fill(Rivet::deltaPhi(jet1.phi(), z.phi()), weight);
    }
  }

  // Reconstruct Z
  const ZFinder& zfinder = applyProjection<ZFinder>(event, "ZFinder");
  if (zfinder.bosons().size() < 1) continue;

  const Particle & z = zfinder.bosons()[0];
  double zpt = z.pt();

  // Now do selection criteria
  bool passZpJ = false;
  if (jets.size() < 1) continue;
  PseudoJet jet1;
  bool overlap = false;
  for (const auto & j : jets) {
    overlap = false;
    for (const auto & m : muons.particlesByPt()) {
      if ((m.pt() > 0.5*j.pt()) && (j.squared_distance(m) < jetRadius*jetRadius)) {
        overlap = true; break; }
    }
    if (!overlap) { jet1 = j; break; } // Jet without overlap found
  }
  if (overlap) continue; // No jet without overlap found
  float jet1pt = jet1.pt();
  float asym = fabs((jet1pt - zpt) / (jet1pt+zpt));
  float dphi = Rivet::deltaPhi(jet1.phi(), z.phi());
  passZpJ = ((zpt > 30) && (asym < 0.3) && (dphi > 2.0));

  if (!passZpJ) continue;

  // Now calculate lambda variables and fill hists
  // _h_pt_jet->fill(jet1pt, weight);

  // Simplify life - ignore this jet if it is below 1st hist pt range
  // Note that we don't apply it to the original jet pt cut - since
  // we have phase space where one jet is > 50, and one < 50
  if (jet1pt < _ptBinsGen[0]) continue;
  // ignore jet if beyond the last bin
  if (jet1pt > _ptBinsGen.back()) continue;
```

```
    const Particle & muon1 = z.constituents()[0];
    const Particle & muon2 = z.constituents()[1];
    _h_pt_z->fill(zpt, weight);
    _h_pt_mu->fill(muon1.pt(), weight);
    _h_pt_mu->fill(muon2.pt(), weight);
    _h_eta_mu->fill(muon1.eta(), weight);
    _h_eta_mu->fill(muon2.eta(), weight);
    _h_eta_jet->fill(jet1.rapidity(), weight);
    _h_dphi_jet_z->fill(dphi, weight);

    // Need to use original, ungroomed jet pT to bin
    uint ptBinInd = getBinIndex(jet1pt, _ptBinsGen);

    // UNGROOMED VERSION
    // ----------------------------------------------------------------

    // Fill hists for each lambda variable
    for (uint lambdaInd=0; lambdaInd < _lambdaVars.size(); lambdaInd++) {
      const LambdaVar & thisLambdaVar = _lambdaVars[lambdaInd];
      Angularity angularity(thisLambdaVar.beta, jetRadius, thisLambdaVar.kappa, thisLambdaVar.isCharged, thisLambdaVar.
          constitCut);
      float val = angularity(jet1);
      if (val<0) continue;

      _h_zpj[radiusInd][lambdaInd][ptBinInd]->fill(val, weight);
    }

    // GROOMED VERSION
    // ----------------------------------------------------------------
    // Get groomed jet
    fastjet::contrib::SoftDrop sd(0, 0.1, jetRadius);
    PseudoJet groomedJet = sd(jet1);
    //fastjet::contrib::ModifiedMassDropTagger mmdt(0.1); mmdt.set_grooming_mode(); mmdt.set_reclustering(false);
    //Recluster ca_cluster(JetDefinition(cambridge_algorithm, JetDefinition::max_allowable_R));
    //PseudoJet groomedJet = mmdt(ca_cluster(jetItr));

    // Fill hists for each lambda variable
    for (uint lambdaInd=0; lambdaInd < _lambdaVars.size(); lambdaInd++) {
      const LambdaVar & thisLambdaVar = _lambdaVars[lambdaInd];
      Angularity angularity(thisLambdaVar.beta, jetRadius, thisLambdaVar.kappa, thisLambdaVar.isCharged, thisLambdaVar.
          constitCut);
      float val = angularity(groomedJet);
      if (val<0) continue;

      _h_zpj_groomed[radiusInd][lambdaInd][ptBinInd]->fill(val, weight);
    }
  } // end loop over jet radii
}

/// Normalise histograms etc., after the run
void finalize() {

  // norm hists to cross section
  // scale(_h_pt_jet, lumi * crossSection() / sumOfWeights());

} // end of finalize

// Order matters here
const vector<float> _jetRadii = {0.4, 0.8};

// This order is important! index in vector used to create YODA plot name
// Must match that in extracRivetPlotsZPJ.py
const std::vector<LambdaVar> _lambdaVars = {
  LambdaVar("jet_multiplicity", 0, 0, false, SelectorPtMin(1.)),
  LambdaVar("jet_pTD", 2, 0, false, SelectorPtMin(0.)),
  LambdaVar("jet_LHA", 1, 0.5, false, SelectorPtMin(0.)),
  LambdaVar("jet_width", 1, 1, false, SelectorPtMin(0.)),
  LambdaVar("jet_thrust", 1, 2, false, SelectorPtMin(0.)),
  LambdaVar("jet_multiplicity_charged", 0, 0, true, SelectorPtMin(1.)),
  LambdaVar("jet_pTD_charged", 2, 0, true, SelectorPtMin(0.)),
  LambdaVar("jet_LHA_charged", 1, 0.5, true, SelectorPtMin(0.)),
  LambdaVar("jet_width_charged", 1, 1, true, SelectorPtMin(0.)),
  LambdaVar("jet_thrust_charged", 1, 2, true, SelectorPtMin(0.)),
```

```
    };

    const std::vector<float> _ptBinsGen = {
      50, 65, 88, 120, 150, 186, 254, 326, 408, 481, 614, 800, 2000
    };

    /// @name Histograms
    Histo1DPtr _h_n_muons_no_cut;
    Histo1DPtr _h_pt_z_no_cut;
    Histo1DPtr _h_pt_jet_no_cut, _h_eta_jet_no_cut;
    Histo1DPtr _h_pt_mu_no_cut, _h_eta_mu_no_cut;
    Histo1DPtr _h_dphi_jet_z_no_cut;
    Histo1DPtr _h_pt_z;
    Histo1DPtr _h_pt_jet, _h_eta_jet;
    Histo1DPtr _h_pt_mu, _h_eta_mu;
    Histo1DPtr _h_dphi_jet_z;

    // 3D vector: [jet radius][lambda variable][pt bin]
    // since each pt bin has its own normalised distribution
    vector<vector<vector<Histo1DPtr> > > _h_zpj, _h_zpj_groomed;

  };


  // The hook for the plugin system
  DECLARE_RIVET_PLUGIN(CMS_2018_PAS_SMP_18_QGX_ZPJ);


}
```

Appendix C.

Configuration Files

C.1. Herwig7 CH2

```
# Auto generated configuration file
# using:
# Revision: 1.19
# Source: /local/reps/CMSSW/CMSSW/Configuration/Applications/python/ConfigBuilder.py,v
# with command line options: Configuration/GenProduction/python/ThirteenTeV/QCD_Pt-15To7000_TuneCUETP8M1_Flat_13TeV-
    pythia8_cff.py -s GEN --datatier=GEN-SIM --conditions auto:mc --eventcontent RAWSIM --no_exec -n 10000 --
    python_filename=RIVET_QCD_Pt-15To7000_TuneCUETP8M1_Flat_13TeV-pythia8_cff.py --customise=Configuration/GenProduction/
    rivet_customize.py

# NB: run inside CMSSW_7_1_19 to recover original 2016 sample

import FWCore.ParameterSet.Config as cms

from Configuration.StandardSequences.Eras import eras

process = cms.Process('GEN')

# import of standard configurations
process.load('Configuration.StandardSequences.Services_cff')
process.load('SimGeneral.HepPDTESSource.pythiapdt_cfi')
process.load('FWCore.MessageService.MessageLogger_cfi')
process.load('Configuration.EventContent.EventContent_cff')
process.load('SimGeneral.MixingModule.mixNoPU_cfi')
process.load('Configuration.StandardSequences.GeometryRecoDB_cff')
process.load('Configuration.StandardSequences.MagneticField_cff')
process.load('Configuration.StandardSequences.Generator_cff')
process.load('IOMC.EventVertexGenerators.VtxSmearedRealistic50ns13TeVCollision_cfi')
process.load('GeneratorInterface.Core.genFilterSummary_cff')
process.load('Configuration.StandardSequences.EndOfProcess_cff')
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cff')

process.maxEvents = cms.untracked.PSet(
    input = cms.untracked.int32(10000)
)

# Input source
process.source = cms.Source("EmptySource")

process.options = cms.untracked.PSet(

)

process.MessageLogger.cerr.FwkReport.reportEvery = 1000

# Production Info
process.configurationMetadata = cms.untracked.PSet(
    annotation = cms.untracked.string('Configuration/GenProduction/python/ThirteenTeV/QCD_Pt-15
        To7000_TuneCUETP8M1_Flat_13TeV-pythia8_cff.py nevts:10000'),
    name = cms.untracked.string('Applications'),
    version = cms.untracked.string('$Revision: 1.19 $')
)
```

```
# Output definition

process.RAWSIMoutput = cms.OutputModule("PoolOutputModule",
    SelectEvents = cms.untracked.PSet(
        SelectEvents = cms.vstring('generation_step')
    ),
    compressionAlgorithm = cms.untracked.string('LZMA'),
    compressionLevel = cms.untracked.int32(9),
    dataset = cms.untracked.PSet(
        dataTier = cms.untracked.string('GEN-SIM'),
        filterName = cms.untracked.string('')
    ),
    eventAutoFlushCompressedSize = cms.untracked.int32(20971520),
    fileName = cms.untracked.string('QCD_Pt-15To7000_TuneCUETP8M1_Flat_13TeV-pythia8_cff_py_GEN.root'),
    outputCommands = process.RAWSIMEventContent.outputCommands,
    splitLevel = cms.untracked.int32(0)
)

# Additional output definition

# Other statements
process.genstepfilter.triggerConditions=cms.vstring("generation_step")
from Configuration.AlCa.GlobalTag import GlobalTag
process.GlobalTag = GlobalTag(process.GlobalTag, 'auto:mc', '')

herwig7CH2SettingsBlock = cms.PSet(
    herwig7CH2PDF = cms.vstring(
            'cd /Herwig/Partons',
            'create ThePEG::LHAPDF PDFSet_nnlo ThePEGLHAPDF.so',
            'set PDFSet_nnlo:PDFName NNPDF31_nnlo_as_0118.LHgrid',
            'set PDFSet_nnlo:RemnantHandler HadronRemnants',
            'set /Herwig/Particles/p+:PDF PDFSet_nnlo',
            'set /Herwig/Particles/pbar-:PDF PDFSet_nnlo',

            'set /Herwig/Partons/PPExtractor:FirstPDF  PDFSet_nnlo',
            'set /Herwig/Partons/PPExtractor:SecondPDF PDFSet_nnlo',

            'set /Herwig/Shower/ShowerHandler:PDFA PDFSet_nnlo',
            'set /Herwig/Shower/ShowerHandler:PDFB PDFSet_nnlo',

            'create ThePEG::LHAPDF PDFSet_lo ThePEGLHAPDF.so',
            ## difference to CH3
            'set PDFSet_lo:PDFName NNPDF31_lo_as_0118.LHgrid',
            'set PDFSet_lo:RemnantHandler HadronRemnants',

            'set /Herwig/Shower/ShowerHandler:PDFARemnant PDFSet_lo',
            'set /Herwig/Shower/ShowerHandler:PDFBRemnant PDFSet_lo',
            'set /Herwig/Partons/MPIExtractor:FirstPDF PDFSet_lo',
            'set /Herwig/Partons/MPIExtractor:SecondPDF PDFSet_lo',

            'cd /',
        ),
    herwig7CH2AlphaS = cms.vstring(
        'cd /Herwig/Shower',
        'set AlphaQCD:AlphaMZ 0.136',
        'cd /'
        ),
    herwig7CH2MPISettings = cms.vstring(
        'read snippets/SoftModel.in',
        ## difference to CH3
        'set /Herwig/Hadronization/ColourReconnector:ReconnectionProbability 0.479',
        'set /Herwig/UnderlyingEvent/MPIHandler:pTmin0 3.138',
        'set /Herwig/UnderlyingEvent/MPIHandler:InvRadius 1.174',
        'set /Herwig/UnderlyingEvent/MPIHandler:Power 0.1203',
        'set /Herwig/Partons/RemnantDecayer:ladderPower -0.08',
        'set /Herwig/Partons/RemnantDecayer:ladderNorm 0.95',
                                )
)

# from Configuration.Generator.Herwig7Settings.Herwig7CH2TuneSettings_cfi import herwig7CH2SettingsBlock
process.generator = cms.EDFilter("Herwig7GeneratorFilter",
    herwig7CH2SettingsBlock,
    productionParameters = cms.vstring(
```

```
        'read snippets/PPCollider.in',
        ## UE
        'read snippets/MB.in',
        'read snippets/Diffraction.in',
        ## UE ends
        'mkdir /Herwig/Weights',
        'cd /Herwig/Weights',
        'create ThePEG::ReweightMinPT reweightMinPT ReweightMinPT.so',
        'cd /Herwig/MatrixElements/',
        ## DIJET
        # 'insert SubProcess:MatrixElements[0] MEQCD2to2',
        # 'insert SubProcess:Preweights[0] /Herwig/Weights/reweightMinPT',
        ## DIJET ends
        ## ZPJ
        # 'insert SubProcess:MatrixElements[0] MEZJet',
        # 'DISABLEREADONLY',
        # 'newdef MEZJet:ZDecay ChargedLeptons',
        ## ZPJ ends
        'cd /',
        ## ZPJ
        # 'set /Herwig/Cuts/LeptonPairMassCut:MinMass 60.*GeV',
        ## ZPJ ends
        'set /Herwig/Cuts/JetKtCut:MinKT 15.*GeV',
        'set /Herwig/Cuts/JetKtCut:MaxKT 7000.*GeV',
        'set /Herwig/Cuts/Cuts:MHatMin 0.0*GeV',
        'set /Herwig/Cuts/Cuts:X1Min 1e-07',
        'set /Herwig/Cuts/Cuts:X2Min 1e-07',
        'set /Herwig/Cuts/MassCut:MinM 0.0*GeV',
        'set /Herwig/Weights/reweightMinPT:Power 4.5',
        'set /Herwig/Weights/reweightMinPT:Scale 15*GeV',
        'set /Herwig/Decays/DecayHandler:MaxLifeTime 10*mm',
        'set /Herwig/Decays/DecayHandler:LifeTimeOption Average',
    ),
    parameterSets = cms.vstring('productionParameters', 'herwig7CH2PDF','herwig7CH2AlphaS', 'herwig7CH2MPISettings'),
    configFiles = cms.vstring(),
    crossSection = cms.untracked.double(1363000000),
    dataLocation = cms.string('${HERWIGPATH:-6}'),
    eventHandlers = cms.string('/Herwig/EventHandlers'),
    filterEfficiency = cms.untracked.double(1.0),
    generatorModule = cms.string('/Herwig/Generators/EventGenerator'),
    repository = cms.string('${HERWIGPATH}/HerwigDefaults.rpo'),
    run = cms.string('InterfaceMatchboxTest'),
    runModeList = cms.untracked.string("read,run"),
)

# Set random number seed
# process.RandomNumberGeneratorService.generator.initialSeed = cms.untracked.uint32(123456789)

# Path and EndPath definitions
process.generation_step = cms.Path(process.pgen)
process.genfiltersummary_step = cms.EndPath(process.genFilterSummary)
process.endjob_step = cms.EndPath(process.endOfProcess)
process.RAWSIMoutput_step = cms.EndPath(process.RAWSIMoutput)

# Schedule definition
process.schedule = cms.Schedule(process.generation_step,process.genfiltersummary_step,process.endjob_step,process.
    RAWSIMoutput_step)
from PhysicsTools.PatAlgos.tools.helpers import associatePatAlgosToolsTask
associatePatAlgosToolsTask(process)
# filter all path with the production filter sequence
for path in process.paths:
    getattr(process,path)._seq = process.generator * getattr(process,path)._seq

# customisation of the process.

# Automatic addition of the customisation function from Configuration.GenProduction.rivet_customize
# from Configuration.GenProduction.rivet_customize import customise

#call to customisation function customise imported from Configuration.GenProduction.rivet_customize
# process = customise(process)

process.load('GeneratorInterface.RivetInterface.rivetAnalyzer_cfi')
process.generation_step += process.rivetAnalyzer
process.schedule.remove(process.RAWSIMoutput_step)
```

```
process.rivetAnalyzer.CrossSection = cms.double(7.782e+10)
## DIJET
##  AlphaQCD:AlphaMZ
##  default    0.113       0.123
##  1.384e+09  1.312e+09   1.349e+09
## ZPJ
##  AlphaQCD:AlphaMZ
##  default    0.113       0.123
##  1.605e+03  1.597e+03   1.612e+03
## UE
##  AlphaQCD:AlphaMZ
##  default    0.113       0.123      0.100      0.136
##  7.782e+10  7.781e+10   7.782e+10  7.782e+10  7.782e+10

process.rivetAnalyzer.UseExternalWeight = cms.bool(True)  # for weighted events
process.rivetAnalyzer.useGENweights = cms.bool(True)
process.rivetAnalyzer.useLHEweights = cms.bool(False)  # doesn't matter as no separate LHE generator

process.rivetAnalyzer.AnalysisNames = cms.vstring('CMS_Internal_FSQ_15_007','CMS_FSQ_15_008','CMS_2015_I1384119','
    ATLAS_2016_I1419652','CMS_FSQ_15_005','CMS_FSQ_15_006')
## Substructure
## CMS_2018_PAS_SMP_18_QGX_ZPJ
## CMS_2018_PAS_SMP_18_QGX_DIJET
## Underlying events
## CMS_Internal_FSQ_15_007','CMS_FSQ_15_008','CMS_2015_I1384119','ATLAS_2016_I1419652','CMS_FSQ_15_005','CMS_FSQ_15_006
process.rivetAnalyzer.OutputFile = cms.string('H7_CH2_UE_AlphaS_0136.yoda')

# End of customisation functions

# Customisation from command line

# Add early deletion of temporary data products to reduce peak memory need
from Configuration.StandardSequences.earlyDeleteSettings_cff import customiseEarlyDelete
process = customiseEarlyDelete(process)
# End adding early deletion

print("Running with seed", process.RandomNumberGeneratorService.generator.initialSeed)
```

C.2. Pythia8 CR0

```
import FWCore.ParameterSet.Config as cms
import os

process = cms.Process('GEN')

# import of standard configurations
process.load('Configuration.StandardSequences.Services_cff')
process.load('SimGeneral.HepPDTESSource.pythiapdt_cfi')
process.load('FWCore.MessageService.MessageLogger_cfi')
process.load('Configuration.EventContent.EventContent_cff')
process.load('SimGeneral.MixingModule.mixNoPU_cfi')
process.load('Configuration.StandardSequences.GeometryDB_cff')
process.load('Configuration.StandardSequences.MagneticField_38T_cff')
process.load('Configuration.StandardSequences.Generator_cff')
process.load('IOMC.EventVertexGenerators.VtxSmearedRealistic7TeV2011Collision_cfi')
process.load('GeneratorInterface.Core.genFilterSummary_cff')
process.load('Configuration.StandardSequences.EndOfProcess_cff')
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cff')

process.maxEvents = cms.untracked.PSet(
    input = cms.untracked.int32(10000)
)

# Input source
process.source = cms.Source("EmptySource")

process.options = cms.untracked.PSet(

)
```

```
# Production Info
process.configurationMetadata = cms.untracked.PSet(
    version = cms.untracked.string('$Revision: 1.3 $'),
    annotation = cms.untracked.string('PYTHIA6-MinBias TuneZ2 at 7TeV'),
    name = cms.untracked.string('$Source: /local/reps/CMSSW/CMSSW/Configuration/GenProduction/python/Attic/
        MinBias_TuneZ2_7TeV_pythia6_cff.py,v $')
)

# Output definition

process.RAWSIMoutput = cms.OutputModule("PoolOutputModule",
    splitLevel = cms.untracked.int32(0),
    eventAutoFlushCompressedSize = cms.untracked.int32(5242880),
    outputCommands = process.RAWSIMEventContent.outputCommands,
    fileName = cms.untracked.string('MinBias_TuneZ2_7TeV_pythia6_cff_py_GEN.root'),
    dataset = cms.untracked.PSet(
        filterName = cms.untracked.string(''),
        dataTier = cms.untracked.string('GEN-SIM-RAW')
    ),
    SelectEvents = cms.untracked.PSet(
        SelectEvents = cms.vstring('generation_step')
    )
)

# Additional output definition
process.MessageLogger.cerr.FwkReport.reportEvery = 10000

# Other statements
process.GlobalTag.globaltag = 'MCRUN2_74_V9A'

pythia8CommonSettingsBlock = cms.PSet(
    pythia8CommonSettings = cms.vstring(
      'Tune:preferLHAPDF = 2',
      'Main:timesAllowErrors = 10000',
      'Check:epTolErr = 0.01',
      'Beams:setProductionScalesFromLHEF = off',
      'SLHA:keepSM = on',
      'SLHA:minMassSM = 1000.',
      'ParticleDecays:limitTau0 = on',
      'ParticleDecays:tau0Max = 10',
      'ParticleDecays:allowPhotonRadiation = on',
    )
)

pythia8CP5SettingsBlock = cms.PSet(
    pythia8CP5Settings = cms.vstring(
        'Tune:pp 14',
        'Tune:ee 7',
        'MultipartonInteractions:ecmPow=0.03344',
        'MultipartonInteractions:bProfile=2',
        'MultipartonInteractions:pT0Ref=1.41',
        'MultipartonInteractions:coreRadius=0.7634',
        'MultipartonInteractions:coreFraction=0.63',
        'ColourReconnection:mode=0', # 0
        'ColourReconnection:range=5.176',
        'SigmaTotal:zeroAXB=off',
        'SpaceShower:alphaSorder=2',
        'SpaceShower:alphaSvalue=0.118', # 0.118
        'SigmaProcess:alphaSvalue=0.118',
        'SigmaProcess:alphaSorder=2',
        'MultipartonInteractions:alphaSvalue=0.118',
        'MultipartonInteractions:alphaSorder=2',
        'TimeShower:alphaSorder=2',
        'TimeShower:alphaSvalue=0.118', # 0.118
##  0.118   0.113   0.123
        'TimeShower:pTmin=0.5', # 0.5 from Monash 2013
##  0.5   0.45   0.55   0.2   0.8
        'SigmaTotal:mode = 0',
        'SigmaTotal:sigmaEl = 21.89',
        'SigmaTotal:sigmaTot = 100.309',
        'PDF:pSet=LHAPDF6:NNPDF31_nnlo_as_0118',
        #'TimeShower:scaleGluonToQuark=<__TimeShower:scaleGluonToQuark__>', # 1 Pythia default
        )
)
```

```
process.generator = cms.EDFilter("Pythia8GeneratorFilter",
        comEnergy = cms.double(13000.0),
        crossSection = cms.untracked.double(1.0),
        filterEfficiency = cms.untracked.double(1.0),
        maxEventsToPrint = cms.untracked.int32(1),
        pythiaHepMCVerbosity = cms.untracked.bool(False),
        pythiaPylistVerbosity = cms.untracked.int32(1),

        PythiaParameters = cms.PSet(
                pythia8CommonSettingsBlock,
                pythia8CP5SettingsBlock,
                processParameters = cms.vstring(
                        ## switch on for DIJET
                        # 'HardQCD:all = on',
                        ## switch on for ZPJ
                        # 'WeakBosonAndParton:qqbar2gmZg = on',
                        # 'WeakBosonAndParton:qg2gmZq = on',
                        # 'WeakBosonAndParton:fgm2gmZf = on',
                        # '23:onMode = off',
                        # '23:onIfAny = 13',
                        ## switch on for UE
                        'SoftQCD:nonDiffractive = on',
                        'SoftQCD:singleDiffractive = on',
                        'SoftQCD:centralDiffractive = on',
                        'SoftQCD:doubleDiffractive = on',
                        ## switch on for DIJET and ZPJ
                        # 'PhaseSpace:pTHatMin = 10',
                        # 'PhaseSpace:pTHatMax = 4000',
                        # 'PhaseSpace:bias2Selection = on',
                        # 'PhaseSpace:bias2SelectionPow = 4.5',
                        # 'PhaseSpace:bias2SelectionRef = 15.',
                ),
                parameterSets = cms.vstring('pythia8CommonSettings',
                                            'pythia8CP5Settings',
                                            'processParameters',
                                            )
        )
)

process.ProductionFilterSequence = cms.Sequence(process.generator)

# Path and EndPath definitions
process.generation_step = cms.Path(process.pgen)
process.genfiltersummary_step = cms.EndPath(process.genFilterSummary)
process.endjob_step = cms.EndPath(process.endOfProcess)
process.RAWSIMoutput_step = cms.EndPath(process.RAWSIMoutput)

# Schedule definition
process.schedule = cms.Schedule(process.generation_step,process.genfiltersummary_step,process.endjob_step,process.
    RAWSIMoutput_step)
# filter all path with the production filter sequence
for path in process.paths:
        getattr(process,path)._seq = process.ProductionFilterSequence * getattr(process,path)._seq

# customisation of the process.
process.load('GeneratorInterface.RivetInterface.rivetAnalyzer_cfi')

# Automatic addition of the customisation function from Configuration.GenProduction.rivet_customize
#from Configuration.GenProduction.rivet_customize import customise

def customise(process):
        process.load('GeneratorInterface.RivetInterface.rivetAnalyzer_cfi')
        process.rivetAnalyzer.AnalysisNames = cms.vstring('CMS_Internal_FSQ_15_007','CMS_FSQ_15_008','CMS_2015_I1384119','
            ATLAS_2016_I1419652','CMS_FSQ_15_005','CMS_FSQ_15_006')
## DIJET
## CMS_2018_PAS_SMP_18_QGX_DIJET
## ZPJ
## CMS_2018_PAS_SMP_18_QGX_ZPJ
## UE
## CMS_Internal_FSQ_15_007','CMS_FSQ_15_008','CMS_2015_I1384119','ATLAS_2016_I1419652','CMS_FSQ_15_005','CMS_FSQ_15_006

        process.rivetAnalyzer.CrossSection = cms.double(7.842e+10)
```

```
##  DIJET
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123
##  4.956e+09   4.948e+09   4.937e+09
##
##  ZPJ
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123
##  1.339e+03   1.333e+03   1.338e+03
##
##  Underlying Events
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123       0.100       0.136
##  7.842e+10   7.842e+10   7.842e+10   7.842e+10   7.842e+10

##  ALL 4 ALPHA_S-VALUES CHANGED
##  DIJET
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123       0.100       0.136
##  4.956e+09   4.374e+09   5.622e+09   3.147e+09   7.576e+09
##
##  ZPJ
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123       0.100       0.136
##  1.339e+03   1.201e+03   1.401e+03   1.100e+03   1.584e+03
##
##  Underlying Events
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123       0.100       0.136
##  7.842e+10   7.842e+10   7.842e+10   7.842e+10   7.842e+10

        process.rivetAnalyzer.OutputFile = cms.string('P8_CR0_UE_all4AlphaS_0118.yoda')
        process.rivetAnalyzer.UseExternalWeight = cms.bool(True)
        process.rivetAnalyzer.useGENweights = cms.bool(True)
        process.generation_step+=process.rivetAnalyzer
        process.schedule.remove(process.RAWSIMoutput_step)
        return(process)
#call to customisation function customise imported from Configuration.GenProduction.rivet_customize
process = customise(process)

# End of customisation functions
```

C.3. Pythia8 CR1

```
import FWCore.ParameterSet.Config as cms
import os

process = cms.Process('GEN')

# import of standard configurations
process.load('Configuration.StandardSequences.Services_cff')
process.load('SimGeneral.HepPDTESSource.pythiapdt_cfi')
process.load('FWCore.MessageService.MessageLogger_cfi')
process.load('Configuration.EventContent.EventContent_cff')
process.load('SimGeneral.MixingModule.mixNoPU_cfi')
process.load('Configuration.StandardSequences.GeometryDB_cff')
process.load('Configuration.StandardSequences.MagneticField_38T_cff')
process.load('Configuration.StandardSequences.Generator_cff')
process.load('IOMC.EventVertexGenerators.VtxSmearedRealistic7TeV2011Collision_cfi')
process.load('GeneratorInterface.Core.genFilterSummary_cff')
process.load('Configuration.StandardSequences.EndOfProcess_cff')
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cff')

process.maxEvents = cms.untracked.PSet(
    input = cms.untracked.int32(10000)
)

# Input source
process.source = cms.Source("EmptySource")

process.options = cms.untracked.PSet(
```

```
)

# Production Info
process.configurationMetadata = cms.untracked.PSet(
    version = cms.untracked.string('$Revision: 1.3 $'),
    annotation = cms.untracked.string('PYTHIA6-MinBias TuneZ2 at 7TeV'),
    name = cms.untracked.string('$Source: /local/reps/CMSSW/CMSSW/Configuration/GenProduction/python/Attic/
        MinBias_TuneZ2_7TeV_pythia6_cff.py,v $')
)

# Output definition

process.RAWSIMoutput = cms.OutputModule("PoolOutputModule",
    splitLevel = cms.untracked.int32(0),
    eventAutoFlushCompressedSize = cms.untracked.int32(5242880),
    outputCommands = process.RAWSIMEventContent.outputCommands,
    fileName = cms.untracked.string('MinBias_TuneZ2_7TeV_pythia6_cff_py_GEN.root'),
    dataset = cms.untracked.PSet(
        filterName = cms.untracked.string(''),
        dataTier = cms.untracked.string('GEN-SIM-RAW')
    ),
    SelectEvents = cms.untracked.PSet(
        SelectEvents = cms.vstring('generation_step')
    )
)

# Additional output definition
process.MessageLogger.cerr.FwkReport.reportEvery = 10000

# Other statements
process.GlobalTag.globaltag = 'MCRUN2_74_V9A'

pythia8CommonSettingsBlock = cms.PSet(
    pythia8CommonSettings = cms.vstring(
      'Tune:preferLHAPDF = 2',
      'Main:timesAllowErrors = 10000',
      'Check:epTolErr = 0.01',
      'Beams:setProductionScalesFromLHEF = off',
      'SLHA:keepSM = on',
      'SLHA:minMassSM = 1000.',
      'ParticleDecays:limitTau0 = on',
      'ParticleDecays:tau0Max = 10',
      'ParticleDecays:allowPhotonRadiation = on',
    )
)

pythia8CP5SettingsBlock = cms.PSet(
    pythia8CP5Settings = cms.vstring(
        'Tune:pp 14',
        'Tune:ee 7',
        'MultipartonInteractions:ecmPow=0.03283',
        'MultipartonInteractions:bProfile=2',
        'MultipartonInteractions:pT0Ref=1.375',
        'MultipartonInteractions:coreRadius=0.6046',
        'MultipartonInteractions:coreFraction=0.4446',
        'ColourReconnection:mode=1',
        'BeamRemnants:remnantMode=1',
        'ColourReconnection:junctionCorrection=0.238',
        'ColourReconnection:timeDilationPar=8.58',
        'ColourReconnection:m0=1.721',
        'StringZ:aLund=0.38',
        'StringZ:bLund=0.64',
        'StringFlav:probQQtoQ=0.078',
        'StringFlav:probStoUD=0.2',
        'SpaceShower:alphaSorder=2',
        'SpaceShower:alphaSvalue=0.118', # 0.118
        'SigmaProcess:alphaSvalue=0.118',
        'SigmaProcess:alphaSorder=2',
        'MultipartonInteractions:alphaSvalue=0.118',
        'MultipartonInteractions:alphaSorder=2',
        'TimeShower:alphaSorder=2',
        'TimeShower:alphaSvalue=0.136', # 0.118
##   0.118   0.113   0.123
```

```
        'TimeShower:pTmin=0.5', # 0.5 from Monash 2013
##  0.5   0.45   0.55   0.2   0.8
        'SigmaTotal:mode = 0',
        'SigmaTotal:sigmaEl = 21.89',
        'SigmaTotal:sigmaTot = 100.309',
        'PDF:pSet=LHAPDF6:NNPDF31_nnlo_as_0118',
        )
)

process.generator = cms.EDFilter("Pythia8GeneratorFilter",
        comEnergy = cms.double(13000.0),
        crossSection = cms.untracked.double(1.0),
        filterEfficiency = cms.untracked.double(1.0),
        maxEventsToPrint = cms.untracked.int32(1),
        pythiaHepMCVerbosity = cms.untracked.bool(False),
        pythiaPylistVerbosity = cms.untracked.int32(1),

        PythiaParameters = cms.PSet(
                pythia8CommonSettingsBlock,
                pythia8CP5SettingsBlock,
                processParameters = cms.vstring(
                        ## switch on for DIJET
                        # 'HardQCD:all = on',
                        ## switch on for ZPJ
                        # 'WeakBosonAndParton:qqbar2gmZg = on',
                        # 'WeakBosonAndParton:qg2gmZq = on',
                        # 'WeakBosonAndParton:fgm2gmZf = on',
                        # '23:onMode = off',
                        # '23:onIfAny = 13',
                        ## switch on for UE
                        'SoftQCD:nonDiffractive = on',
                        'SoftQCD:singleDiffractive = on',
                        'SoftQCD:centralDiffractive = on',
                        'SoftQCD:doubleDiffractive = on',
                        ## switch on for DIJET and ZPJ
                        # 'PhaseSpace:pTHatMin = 10',
                        # 'PhaseSpace:pTHatMax = 4000',
                        # 'PhaseSpace:bias2Selection = on',
                        # 'PhaseSpace:bias2SelectionPow = 4.5',
                        # 'PhaseSpace:bias2SelectionRef = 15.',
                ),
                parameterSets = cms.vstring('pythia8CommonSettings',
                                            'pythia8CP5Settings',
                                            'processParameters',
                                            )
        )
)

process.ProductionFilterSequence = cms.Sequence(process.generator)

# Path and EndPath definitions
process.generation_step = cms.Path(process.pgen)
process.genfiltersummary_step = cms.EndPath(process.genFilterSummary)
process.endjob_step = cms.EndPath(process.endOfProcess)
process.RAWSIMoutput_step = cms.EndPath(process.RAWSIMoutput)

# Schedule definition
process.schedule = cms.Schedule(process.generation_step,process.genfiltersummary_step,process.endjob_step,process.
    RAWSIMoutput_step)
# filter all path with the production filter sequence
for path in process.paths:
        getattr(process,path)._seq = process.ProductionFilterSequence * getattr(process,path)._seq

# customisation of the process.
process.load('GeneratorInterface.RivetInterface.rivetAnalyzer_cfi')

# Automatic addition of the customisation function from Configuration.GenProduction.rivet_customize
#from Configuration.GenProduction.rivet_customize import customise

def customise(process):
        process.load('GeneratorInterface.RivetInterface.rivetAnalyzer_cfi')
        process.rivetAnalyzer.AnalysisNames = cms.vstring('CMS_Internal_FSQ_15_007','CMS_FSQ_15_008','CMS_2015_I1384119','
            ATLAS_2016_I1419652','CMS_FSQ_15_005','CMS_FSQ_15_006')
```

```
## DIJET
## CMS_2018_PAS_SMP_18_QGX_DIJET
## ZPJ
## CMS_2018_PAS_SMP_18_QGX_ZPJ
## UE
## CMS_Internal_FSQ_15_007','CMS_FSQ_15_008','CMS_2015_I1384119','ATLAS_2016_I1419652','CMS_FSQ_15_005','CMS_FSQ_15_006
        process.rivetAnalyzer.CrossSection = cms.double(7.842e+10)

##  DIJET
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123
##  4.986e+09   4.936e+09   4.921e+09
##  TimeShower:pTmin
##  0.5         0.45        0.55        0.2         0.8
##
##
##  ZPJ
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123
##  1.334e+03   1.331e+02   1.339e+03
##
##  UE
##  TimeShower:alphaSvalue
##  0.118       0.113       0.123       0.100       0.136
##  7.842e+10   7.842e+10   7.842e+10   7.842e+10   7.842e+10

        process.rivetAnalyzer.OutputFile = cms.string('P8_CR1_UE_AlphaS_0136.yoda')
        process.rivetAnalyzer.UseExternalWeight = cms.bool(True)
        process.rivetAnalyzer.useGENweights = cms.bool(True)
        process.generation_step+=process.rivetAnalyzer
        process.schedule.remove(process.RAWSIMoutput_step)
        return(process)
#call to customisation function customise imported from Configuration.GenProduction.rivet_customize
process = customise(process)

# End of customisation functions
```

C.4. Pythia8 CR2

```
import FWCore.ParameterSet.Config as cms
import os

process = cms.Process('GEN')

# import of standard configurations
process.load('Configuration.StandardSequences.Services_cff')
process.load('SimGeneral.HepPDTESSource.pythiapdt_cfi')
process.load('FWCore.MessageService.MessageLogger_cfi')
process.load('Configuration.EventContent.EventContent_cff')
process.load('SimGeneral.MixingModule.mixNoPU_cfi')
process.load('Configuration.StandardSequences.GeometryDB_cff')
process.load('Configuration.StandardSequences.MagneticField_38T_cff')
process.load('Configuration.StandardSequences.Generator_cff')
process.load('IOMC.EventVertexGenerators.VtxSmearedRealistic7TeV2011Collision_cfi')
process.load('GeneratorInterface.Core.genFilterSummary_cff')
process.load('Configuration.StandardSequences.EndOfProcess_cff')
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cff')

process.maxEvents = cms.untracked.PSet(
    input = cms.untracked.int32(10000)
)

# Input source
process.source = cms.Source("EmptySource")

process.options = cms.untracked.PSet(

)

# Production Info
process.configurationMetadata = cms.untracked.PSet(
```

```
    version = cms.untracked.string('$Revision: 1.3 $'),
    annotation = cms.untracked.string('PYTHIA6-MinBias TuneZ2 at 7TeV'),
    name = cms.untracked.string('$Source: /local/reps/CMSSW/CMSSW/Configuration/GenProduction/python/Attic/
        MinBias_TuneZ2_7TeV_pythia6_cff.py,v $')
)

# Output definition

process.RAWSIMoutput = cms.OutputModule("PoolOutputModule",
    splitLevel = cms.untracked.int32(0),
    eventAutoFlushCompressedSize = cms.untracked.int32(5242880),
    outputCommands = process.RAWSIMEventContent.outputCommands,
    fileName = cms.untracked.string('MinBias_TuneZ2_7TeV_pythia6_cff_py_GEN.root'),
    dataset = cms.untracked.PSet(
        filterName = cms.untracked.string(''),
        dataTier = cms.untracked.string('GEN-SIM-RAW')
    ),
    SelectEvents = cms.untracked.PSet(
        SelectEvents = cms.vstring('generation_step')
    )
)

# Additional output definition
process.MessageLogger.cerr.FwkReport.reportEvery = 10000

# Other statements
process.GlobalTag.globaltag = 'MCRUN2_74_V9A'

pythia8CommonSettingsBlock = cms.PSet(
    pythia8CommonSettings = cms.vstring(
      'Tune:preferLHAPDF = 2',
      'Main:timesAllowErrors = 10000',
      'Check:epTolErr = 0.01',
      'Beams:setProductionScalesFromLHEF = off',
      'SLHA:keepSM = on',
      'SLHA:minMassSM = 1000.',
      'ParticleDecays:limitTau0 = on',
      'ParticleDecays:tau0Max = 10',
      'ParticleDecays:allowPhotonRadiation = on',
    )
)

pythia8CP5SettingsBlock = cms.PSet(
    pythia8CP5Settings = cms.vstring(
        'Tune:pp 14',
        'Tune:ee 7',
        'MultipartonInteractions:ecmPow=0.0555', #0.05357',
        'MultipartonInteractions:bProfile=2',
        'MultipartonInteractions:pT0Ref=1.454',
        'MultipartonInteractions:coreRadius=0.6532', #0.6489',
        'MultipartonInteractions:coreFraction=0.4392', #0.4881',
        'ColourReconnection:mode=2',
        'ColourReconnection:m2Lambda=4.395', #4.917',
        'ColourReconnection:fracGluon=0.9896', #0.9927',
        'SigmaTotal:zeroAXB=off',
        'SpaceShower:alphaSorder=2',
        'SpaceShower:alphaSvalue=0.118', # 0.118
        'SigmaProcess:alphaSvalue=0.118',
        'SigmaProcess:alphaSorder=2',
        'MultipartonInteractions:alphaSvalue=0.118',
        'MultipartonInteractions:alphaSorder=2',
        'TimeShower:alphaSorder=2',
        'TimeShower:alphaSvalue=0.136', # 0.118
##  0.118   0.113   0.123
        'TimeShower:pTmin=0.5', # 0.5 from Monash 2013
##  0.5   0.45   0.55   0.2   0.8
        'SigmaTotal:mode = 0',
        'SigmaTotal:sigmaEl = 21.89',
        'SigmaTotal:sigmaTot = 100.309',
        'PDF:pSet=LHAPDF6:NNPDF31_nnlo_as_0118',
        )
)

process.generator = cms.EDFilter("Pythia8GeneratorFilter",
```

```
        comEnergy = cms.double(13000.0),
        crossSection = cms.untracked.double(1.0),
        filterEfficiency = cms.untracked.double(1.0),
        maxEventsToPrint = cms.untracked.int32(1),
        pythiaHepMCVerbosity = cms.untracked.bool(False),
        pythiaPylistVerbosity = cms.untracked.int32(1),

        PythiaParameters = cms.PSet(
                pythia8CommonSettingsBlock,
                pythia8CP5SettingsBlock,
                processParameters = cms.vstring(
                        ## switch on for DIJET
                        # 'HardQCD:all = on',
                        ## switch on for ZPJ
                        # 'WeakBosonAndParton:qqbar2gmZg = on',
                        # 'WeakBosonAndParton:qg2gmZq = on',
                        # 'WeakBosonAndParton:fgm2gmZf = on',
                        # '23:onMode = off',
                        # '23:onIfAny = 13',
                        ## switch on for UE
                        'SoftQCD:nonDiffractive = on',
                        'SoftQCD:singleDiffractive = on',
                        'SoftQCD:centralDiffractive = on',
                        'SoftQCD:doubleDiffractive = on',
                        ## switch on for DIJET and ZPJ
                        # 'PhaseSpace:pTHatMin = 10',
                        # 'PhaseSpace:pTHatMax = 4000',
                        # 'PhaseSpace:bias2Selection = on',
                        # 'PhaseSpace:bias2SelectionPow = 4.5',
                        # 'PhaseSpace:bias2SelectionRef = 15.',
                ),
                parameterSets = cms.vstring('pythia8CommonSettings',
                                            'pythia8CP5Settings',
                                            'processParameters',
                                            )
        )
)

process.ProductionFilterSequence = cms.Sequence(process.generator)

# Path and EndPath definitions
process.generation_step = cms.Path(process.pgen)
process.genfiltersummary_step = cms.EndPath(process.genFilterSummary)
process.endjob_step = cms.EndPath(process.endOfProcess)
process.RAWSIMoutput_step = cms.EndPath(process.RAWSIMoutput)

# Schedule definition
process.schedule = cms.Schedule(process.generation_step,process.genfiltersummary_step,process.endjob_step,process.
    RAWSIMoutput_step)
# filter all path with the production filter sequence
for path in process.paths:
        getattr(process,path)._seq = process.ProductionFilterSequence * getattr(process,path)._seq

# customisation of the process.
process.load('GeneratorInterface.RivetInterface.rivetAnalyzer_cfi')

# Automatic addition of the customisation function from Configuration.GenProduction.rivet_customize
#from Configuration.GenProduction.rivet_customize import customise

def customise(process):
        process.load('GeneratorInterface.RivetInterface.rivetAnalyzer_cfi')
        process.rivetAnalyzer.AnalysisNames = cms.vstring('CMS_Internal_FSQ_15_007','CMS_FSQ_15_008','CMS_2015_I1384119','
            ATLAS_2016_I1419652','CMS_FSQ_15_005','CMS_FSQ_15_006')
## DIJET
## CMS_2018_PAS_SMP_18_QGX_DIJET
## ZPJ
## CMS_2018_PAS_SMP_18_QGX_ZPJ
## UE
## CMS_Internal_FSQ_15_007','CMS_FSQ_15_008','CMS_2015_I1384119','ATLAS_2016_I1419652','CMS_FSQ_15_005','CMS_FSQ_15_006

        process.rivetAnalyzer.CrossSection = cms.double(7.842e+10)

##   DIJET
##   TimeShower:alphaSvalue
```

```
##   0.118        0.113        0.123
##   4.923e+09    4.978e+09    4.968e+09
##   TimeShower:pTmin
##   0.5          0.45         0.55         0.2          0.8
##
##
##   ZPJ
##   TimeShower:alphaSvalue
##   0.118        0.113        0.123
##   1.329e+03    1.334e+03    1.337e+03
##
##   UE
##   TimeShower:alphaSvalue
##   0.118        0.113        0.123        0.100        0.136
##   7.842e+10    7.842e+10    7.842e+10    7.842e+10    7.842e+10

        process.rivetAnalyzer.OutputFile = cms.string('P8_CR2_UE_AlphaS_0136.yoda')
        process.rivetAnalyzer.UseExternalWeight = cms.bool(True)
        process.rivetAnalyzer.useGENweights = cms.bool(True)
        process.generation_step+=process.rivetAnalyzer
        process.schedule.remove(process.RAWSIMoutput_step)
        return(process)
#call to customisation function customise imported from Configuration.GenProduction.rivet_customize
process = customise(process)

# End of customisation functions
```

Acknowledgements

I would like to thank everyone assisting me in the completion of my master thesis. At first, I would like to thank Dr. Andreas Hinzmann, who gave me the opportunity of writing my thesis in his group and supervising me through this entire time.
Secondly, I would like to thank Prof. Peter Schleper for agreeing to be the second supervisor for my thesis.
A big thank you goes to Robin and Steffen for answering all of my many, many questions, as well as for the enjoyable time in our office. Another special thanks to Robin for proofreading this thesis.
I also want to thank the rest of Andreas' group and the entire particle physics group for welcoming me. Sadly, the second half of my time was spend in my home office and, therefore, I couldn't spend more time with you.
Last but not least, I would like to thank my family and girlfriend for their moral support during my entire master's degree study.

Declaration

Eidesstattliche Erklärung

Ich versichere, dass ich die beigefügte schriftliche Masterarbeit selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle deutlich als Entlehnung kenntlich gemacht. Dies gilt auch für alle Informationen, die dem Internet oder anderer elektronischer Datensammlungen entnommen wurden. Ich erkläre ferner, dass die von mir angefertigte Masterarbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung im Rahmen meines Studiums war. Die von mir eingereichte schriftliche Fassung entspricht jener auf dem elektronischen Speichermedium.
Ich bin damit einverstanden, dass die Masterarbeit veröffentlicht wird.

Ort, Datum Unterschrift

Bibliography

[1] R. Kogler, B. Nachman, A. Schmidt et al., "Jet substructure at the Large Hadron Collider", *Rev. Mod. Phys.* **91** (Dec, 2019) 045003, doi:10.1103/RevModPhys.91.045003.

[2] P. Gras, S. Höche, D. Kar et al., "Systematics of quark/gluon tagging", *Journal of High Energy Physics* **2017** (Jul, 2017) doi:10.1007/jhep07(2017)091.

[3] A. J. Larkoski, J. Thaler, and W. J. Waalewijn, "Gaining (mutual) information about quark/gluon discrimination", *Journal of High Energy Physics* **2014** (Nov, 2014) doi:10.1007/jhep11(2014)129.

[4] R. C. Aggleton and A. Hinzmann, "Measurement of jet substructure observables of quark and gluon jets (SMP-20-010) [Work in Progress]". http://cms.cern.ch/iCMS/analysisadmin/cadilines?line=SMP-20-010, 2020.

[5] Wikipedia contributors, "Standard Model — Wikipedia, The Free Encyclopedia". https://en.wikipedia.org/w/index.php?title=Standard_Model, 2020. [Online; accessed 2-October-2020].

[6] Wikipedia contributors, "Color confinement — Wikipedia, The Free Encyclopedia". https://en.wikipedia.org/w/index.php?title=Color_confinement, 2020. [Online; accessed 2-October-2020].

[7] Wikipedia contributors, "Weak isospin — Wikipedia, The Free Encyclopedia". https://en.wikipedia.org/w/index.php?title=Weak_isospin, 2020. [Online; accessed 2-October-2020].

[8] Wikipedia contributors, "Antiparticle — Wikipedia, The Free Encyclopedia". https://en.wikipedia.org/w/index.php?title=Antiparticle, 2020. [Online; accessed 2-October-2020].

[9] S. Höche, "Introduction to parton-shower event generators". https://arxiv.org/abs/1411.4085, 2015.

[10] T. Sjöstrand, S. Mrenna, and P. Skands, "PYTHIA 6.4 physics and manual", *Journal of High Energy Physics* **2006** (May, 2006) 026–026, doi:10.1088/1126-6708/2006/05/026.

[11] T. Sjöstrand, S. Ask, J. R. Christiansen et al., "An introduction to PYTHIA 8.2", *Computer Physics Communications* **191** (Jun, 2015) 159–177, doi:10.1016/j.cpc.2015.01.024.

[12] M. Bähr, S. Gieseke, M. A. Gigg et al., "Herwig++ physics and manual", *The European Physical Journal C* **58** (Nov, 2008) 639–707, doi:10.1140/epjc/s10052-008-0798-9.

[13] CMS Collaboration Collaboration, "Extraction and validation of a set of HERWIG 7 tunes from CMS underlying-event measurements", CMS-PAS-GEN-19-001, CERN, Geneva, (2020).

[14] A. M. Sirunyan, A. Tumasyan, W. Adam et al., "Extraction and validation of a new set of CMS pythia8 tunes from underlying-event measurements", *The European Physical Journal C* **80** (Jan, 2020) doi:10.1140/epjc/s10052-019-7499-4.

[15] L. Evans and P. Bryant, "LHC Machine", *Journal of Instrumentation* **3** (aug, 2008) S08001–S08001, doi:10.1088/1748-0221/3/08/s08001.

[16] T. C. Collaboration, S. Chatrchyan, G. Hmayakyan et al., "The CMS experiment at the CERN LHC", *Journal of Instrumentation* **3** (aug, 2008) S08004–S08004, doi:10.1088/1748-0221/3/08/s08004.

[17] F. Marcastel, "CERN's Accelerator Complex. La chaîne des accélérateurs du CERN",. General Photo.

[18] T. A. Collaboration, K. Aamodt, A. A. Quintana et al., "The ALICE experiment at the CERN LHC", *Journal of Instrumentation* **3** (aug, 2008) S08002–S08002, doi:10.1088/1748-0221/3/08/s08002.

[19] T. A. Collaboration, G. Aad, E. Abat et al., "The ATLAS Experiment at the CERN Large Hadron Collider", *Journal of Instrumentation* **3** (aug, 2008) S08003–S08003, doi:10.1088/1748-0221/3/08/s08003.

[20] T. L. Collaboration, A. A. Alves, L. M. A. Filho et al., "The LHCb Detector at the LHC", *Journal of Instrumentation* **3** (aug, 2008) S08005–S08005, doi:10.1088/1748-0221/3/08/s08005.

[21] CMS Collaboration Collaboration, T. Sakuma, "Cutaway diagrams of CMS detector". http://cds.cern.ch/record/2665537, May, 2019.

[22] D. Barney, "CMS Detector Slice". http://cds.cern.ch/record/2120661, Jan, 2016. CMS Collection.

[23] A. Sirunyan, A. Tumasyan, W. Adam et al., "Particle-flow reconstruction and global event description with the CMS detector", *Journal of Instrumentation* **12** (oct, 2017) P10003–P10003, doi:10.1088/1748-0221/12/10/p10003.

[24] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-ktjet clustering algorithm", *Journal of High Energy Physics* **2008** (apr, 2008) 063–063, doi:10.1088/1126-6708/2008/04/063.

[25] CMS Collaboration Collaboration, "Pseudorapidity distribution of charged hadrons in proton-proton collisions at $\sqrt{s} = 13$ TeV", *Phys. Lett. B* **751** (Jul, 2015) 143–163. 21 p, doi:10.1016/j.physletb.2015.10.004. Version published in Phys. Lett. B.

[26] CMS Collaboration Collaboration, "Underlying Event Measurements with Leading Particles and Jets in pp collisions at $\sqrt{s} = 13$ TeV", CMS-PAS-FSQ-15-007, CERN, Geneva, (2015).

[27] C. Bierlich, A. Buckley, J. Butterworth et al., "Robust Independent Validation of Experiment and Theory: Rivet version 3", *SciPost Physics* **8** (Feb, 2020) doi:10.21468/scipostphys.8.2.026.

[28] "PYTHIA 8". http://home.thep.lu.se/~torbjorn/pythia82html/Welcome.html, 2020. [Online; accessed 2-October-2020].

[29] "The Herwig Event Generator". https://herwig.hepforge.org/, 2020. [Online; accessed 2-October-2020].

[30] S. Argyropoulos and T. Sjöstrand, "Effects of color reconnection on $t\bar{t}$ final states at the LHC", *Journal of High Energy Physics* **2014** (Nov, 2014) doi:10.1007/jhep11(2014)043.

[31] J. R. Christiansen and P. Z. Skands, "String formation beyond leading colour", *Journal of High Energy Physics* **2015** (Aug, 2015) doi:10.1007/jhep08(2015)003.

[32] A. Buckley, J. Butterworth, D. Grellscheid et al., "Rivet user manual". https://arxiv.org/abs/1003.0694, 2013.

[33] "Rivet — the particle-physics MC analysis toolkit". https://rivet.hepforge.org/. [Online; accessed 2-October-2020].

List of figures

List of tables