Università degli Studi di Torino

Dipartimento di Fisica

Master Thesis in Nuclear and Subnuclear Physics

Soft QCD parameter tuning using feed-forward neural networks

Relatore:
Prof. Roberto Covarelli

Candidato:
Michael Dessena

Correlatore:
Prof. Marco Monteno

Controrelatore:
Prof. Paolo Torrielli

Anno Accademico 2020/2021

A Federico

"Iniziare non vuol dire che dovrai finire."

RANCORE, S.U.N.S.H.I.N.E.

Abstract

Soft QCD studies require the use of Monte Carlo generators based on phenomenological models: these models introduce a lot of free parameters that have to be tuned with real data. Due to the high computational cost of running a generator it is important to reduce the number of Monte Carlo runs required, this is done employing a parametrization-based approach where the real response of the generator is replaced by a surrogate one. The simplest approach is the use of a polynomial parametrization. In this work, an alternative approach based on machine learning techniques, implemented in the Python package MCNNTUNES by means of Feed-Forward Neural Networks, is used to tune parameters.

This thesis focuses on the analysis of proton-proton collisions obtained thanks to the Large Hadron Collider (LHC) at energy of $\sqrt{s} = 13$ TeV and detected by the CMS experiment. The charged particle tracks of the resulting, complex final state are detected by the CMS inner tracker. In addition to the activity from the main hard scattering, lots of other tracks are collected. This extra activity is called underlying event. The study of the underlying event is important to describe the topology of a real proton-proton collision. The underlying event is studied employing Monte Carlo methods in the transverse regions defined from the products of the main hard scattering. These regions are less affected by the hard interaction effects. The contribution to the parton momentum of the so-called primordial k_T, the transverse motion originating from the Fermi motion of partons inside of hadrons, is also studied within this thesis work. The observables sensitive to this contribution are the p_T and other characteristic distributions of Z-boson production.

Firstly, the basic theoretical concepts used in the Monte Carlo simulations are introduced. The aim of our work was to validate MCNNTUNES as a tool for the tuning of Monte Carlo generators in High Energy Physics simulations. To perform this validation, we try to reproduce an already existing tune for the description of the Underlying Event in hadron-hadron collisions using data collected in Minimum Bias events. So, in the first part the parameters of the PYTHIA8 event generator related to the Multi Parton Interactions and Color Reconnection have been tuned. The tune was performed employing data from the CDF and CMS experiments at different center-of-mass energies. The validation of the MCNNTUNES tool is performed in two steps: a first test with a limited number of free parameters in order to test all the functionalities and check the different operation modes in a simpler case than the actual tune. Once the tool as been tested, the actual tune is performed.

In the second part of the thesis, the MCNNTUNES tool is employed to the tune of the Primordial k_T and Initial State Radiation parameters using data collected in Z boson-production events by the CMS experiment at the center-of-mass energy of 13 TeV. This is the first test of investigating the feasibility of such a tuning in the CMS collaboration.

Contents

Chapter 1

Introduction

High-energy collisions, such as those studied at the Large Hadron Collider (LHC) in Geneva, lead to very complex final states. Usually, hundreds of particles are produced and detected by the experiments with momenta scales ranging over different orders of magnitude. All these observed final states are very complicated to understand and explain. The theory underlying the description of high energy particle interactions is the *standard model of particle* (SM).
The SM is the theory that describes all the elementary particles and the fundamental interactions: electromagnetic, weak and strong[1]. The SM relies on the Quantum Field Theory framework and on the concept of gauge symmetry. The symmetry group for the theory is $SU(3) \times SU(2) \times U(1)$.
The aim of the SM is to describe with a unique formalism all the three interactions. The abelian group $U(1)$ is related to the electromagnetic interaction, the theory based on the gauge group $U(1)$, first introduced by Richard Feynman, is called *quantum electrodynamics* (QED). During the last century, this force was unified to the weak interaction described by the gauge symmetry $SU(2)$ in a single interaction: the Electroweak interaction [1]. Last, the $SU(3)$ group is the one related to the strong interactions of quarks the theory describes the interaction in terms of three possible states of charge called *colors*: red, green and blue. This is the origin of the theory name: *quantum chromodynamics* (QCD).

All the processes in the standard model are evaluated by means of matrix element calculations to give the probability of having an evolution from an initial-state to a defined final-state.
In practice, the matrix elements are too laborious to be evaluated already over the first few orders of perturbation theory, and in particular, the QCD processes have to deal with the intrinsically non-perturbative problem of *color confinement*. Matrix element calculations have to deal with many divergences and often they are performed in an approximated scenario. Then, even if it has been possible to compute these matrix elements correctly, they must be integrated over a large final-state phase space in order to obtain predictions for the description of experimental ob-

[1] Actually, the fundamental interactions are four: there is also the gravitational interaction. The inclusion of a description also for this last interaction is a central problem for modern physics. There are different theories that try to merge the quantum effects with the general relativity that describe the gravitational force but none of these has been able to give a final description yet. This is not important in the framework of the particle physics studies at the collider in fact this force is negligible with respect to the others.

servables. The crucial tool for the description of the processes is the *factorization*, which allows the treatment of processes separating them into different regimes according to the transferred-momentum scale of the interaction. The processes at the highest scale are referred to as *hard processes* and are described in terms of free partons (asymptotic freedom) interacting and generating a small number of energetic outgoing elementary particles. Due to the large momentum transferred in the hard interaction the value of the QCD coupling constant, α_s, is small (as shown in Fig. 1.1), so the perturbative theory is applicable. While, at lower scales, of the order of 1 GeV, the α_s value is large, the QCD become intrinsically non-perturbative and highly interacting, in this scenario the incoming partons are not describable in terms of free partons because they are confined into the beams hadrons[2] and interact in a non-perturbative way to form the observed final-state. These soft processes are the ones that cannot yet be described starting from first principles but have to be modeled by employing phenomenological models based on the introduction of free parameters. These two regimes are connected by an evolutionary process calculable using perturbative QCD. The main consequence of this scale evolution is the so-called *parton showers*: the production of many additional partons in the initial- and final-state.

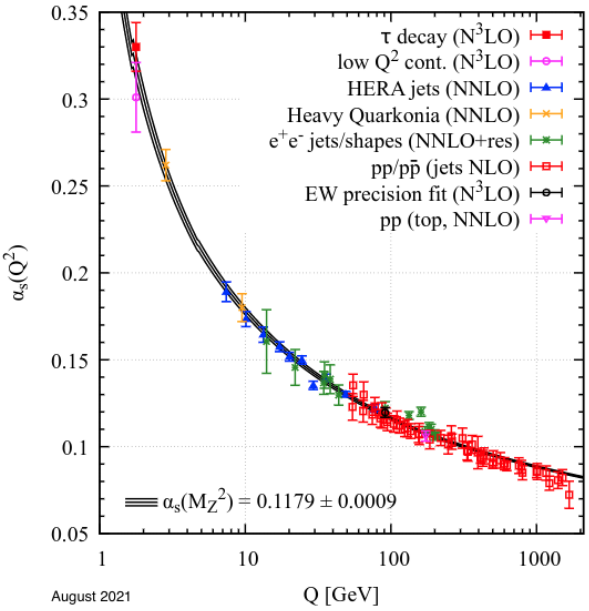


Figure 1.1: Evolution of the QCD coupling constant, α_s, with the energy scale of the interaction [2]

The description of all these processes is based on computer simulation based on Monte Carlo (MC) techniques. The evolution of scale that leads to the parton showering is a Markov process that can be easily simulated with MC tools, also the hadronization that leads from the parton-composed final-state to the hadron-composed final-state is described in terms of MC processes.
The implementation of all these elements together gives a *Monte Carlo event generator* capable of simulating a large number of interesting processes. These event

[2]The coupling constant of the QCD is large so they are strongly bounded to each other and so cannot be seen as free when they interact.

generators are needed in high energy physics studies. For example, they are used to extract a new physics signal from the background of all the other SM processes, provide an input for new selection or reconstruction procedures, or the construction of new experiments.

Historically, the strong interaction was associated with the hadrons and not with quarks (they were unknown yet). However, with the Deep Inelastic Scattering (DIS) experiments [3] it was proven that the hadrons are composite objects and then they cannot be fundamental particles for the SM. There must be something more fundamental: *partons* (quarks and gluons). The *partons model* [4] was developed to describe the interaction of hadrons in very high energy scatterings. What was observed in DIS of electrons on nucleons was the so-called *Bjorken scaling* the observed structure constant was (at high energy) independent from the probe energy, which reflects the fact that the nucleons, or more in general hadrons, are constituted by a collection of point-like objects: the above-mentioned partons.
The development of event generators began shortly after this discovery. The description of lots of processes' features at the hard scale as DIS, hadron jets and lepton pairs production can be performed simply with the description in terms of parton collisions. The description of the complex final-states observed in the MC event generators is obtained by the observation that the primary partons in the evolution can radiate gluons and these gluons could themselves radiate other gluons leading to a cascade. The natural endpoint of this cascade is the *hadronization* that due to the low energy scale and the consequent high value of the coupling constant is not derivable by the QCD fundamental principles.
Most of the interactions observed a the LHC are soft, these soft processes as the above-mentioned hadronization can be simulated but due to the non-perturbative nature of the processes, they must include free parameters that require a *tune* in order to describe the data. A really important phenomenon related to this topic is the study of the so-called *underlying event* (UE) i.e. the soft component of a collision, where at least one hard interaction occurs, that increases the complexity of the final state observed by adding more particles to it.
The hard tail of the UE is described by the QCD perturbative approach using matrix element calculations. But along with it the description of the showering process need to be simulated: the *Initial-State Radiation* (ISR) and *Final-State Radiation* (FSR) add more particles to the simulated process. It is known that the main contribution to the UE arises from the extra partons scatterings occurring along with the primary hard scattering, these extra scatterings are called *Multi Parton Interactions* (MPI). MPI are soft or semi-hard parton-parton scatterings that add more particles to the final-state.
Usually, the observables sensitive to the underlying event receive contribution also from the main hard scattering. So, a good description of UE observables requires a good modeling of MPI and Beam-Beam Remntants (BBR), which are all the partons left unscattered, and also of the hadronization process, ISR and FSR. The existing MC generators, as PYHTIA [5], describe all these processes and have adjustable parameters to control the behavior of event modeling. The set of parameters that best describe the fitted distributions is referred to as *tune*. In 2020 the CMS Collaboration has published a new set of tunes for the UE simulations [6] in PYTHIA8 event generator. The tunes are obtained from data measurements sensitive to soft and

semi-hard MPI at different collision energies.
One of the main problems in the tuning of these event generators is that the tuning procedure can be really computational expensive, due to the high time required to run a MC generator, and to perform a tune it is required to run a generator a very large amount of times. So, to reduce the number of runs required the strategy usually followed for the tuning is to perform a *parameterization-based approach* tuning. The approach consists of fitting the histograms of the observables that are generated from these Monte Carlo simulations with different values for the input parameters under analysis and studying the variation by defining a function that describes the various bins behavior to the variation of the input parameters. These procedure is referred to as *fitting the generator response*.

This master thesis is investigating this tune aspect focusing on the tuning of parameters for the MC generator PYTHIA8. The work is developed within the CMS experiment framework on data collected in minimum bias observation and Z boson production events. In particular, the description of the above mentioned UE. The tuning is performed using MCNNTUNES [7] which implements a machine learning approach by means of Feed Forward Neural Networks (FFNNs). MCNNTUNES wants to be an alternative tool to the tuning problem with respect to the consolidate standard tool, PROFESSOR [8], that implements a parameterization-based approach by using a polynomial function to mimic the generator response.
In the thesis, it is validated MCNNTUNES against the already existing CMS tune for the underlying event: CP5 [6]. The validation is performed employing data from CDF and CMS analyses at different center-of-mass energies: 1.96, 7, 13 TeV and distributions of charged particle density and charged particle transverse momentum sum in the transverse regions and pseudorapidity distributions of charged hadrons production, single and non-single diffractive events. For the description of these distributions a correct tuning for the parameters related to MPI and Color Reconnection (CR) is important.
Once the tool has been validated, it is used to tune the primordial k_T and the ISR cutoff. In this step, it is investigated the effects of the PYTHIA8 parameters related to this two aspects in the description of the low region in the spectra of Drell Yan observation in Z boson production events.

Chapter 2 is going to introduce the theoretical aspects of the description of a hadron-hadron collision. It has to be seen as a general view on all the processes that have to be taken into account to have a proper MC simulation of an hadron-hadron collision. After an overall introduction on the idea behind the MC generators, the chapter starts with the description of factorization theorem that allows to describe hard events using the well-known perturbation theory. This is not only the starting point of the chapter but also of the event simulation in MC generators. Then, the description moves on to the introduction of the parton shower algorithm that links the hard scale of the hard interaction to the softer scale of the hadronization. It has been introduced that the UE description cannot avoid the simulation of the Multi Parton Interaction, so a detailed introduction on this topic is required. The fundamental aspects behind the introduction of the MPI are investigated. It is also described the need of an interleaving between parton shower and MPI algorithms in order to approximate the complexity of the observed real collisions. Some other important processes are described at the parton level, e.g. the introduction of the

primordial k_T and the color reconnection. In the end of the chapter an introduction to the hadronization process is given in particular on its implementation in PYTHIA8 based on the Lund model. Along with the chapter, a particular focus is given to the main parameters that are going to be used in the tuning of the underlying event in minimum bias measurements and in Z boson production spectra observed in Drell Yan events. All the observables used in the tunes presented in this thesis are described in Chapter 3 with a closer look on the CMS detector and on the selection criteria used for the UE analyses. In the last part of the chapter a brief description on the existing CMS tunes called CP and a number from 1 to 5 [6] that are the starting point for this thesis work. Chapter 4 describe the basic concepts in the tuning procedure, in particular, it focuses on the parametrization-based approach used for the tuning, in this chapter, it is also discussed the main tool that will be used called MCNNTUNES [7] which implements a tuning method based on machine learning. The MCNNTUNES tunes performed for the underlying event and for the Z boson spectrum are described respectively in Chapter 5 and Chapter 6.

Chapter 2

Hadron-Hadron Scattering

In a high energy proton-proton collision, it is possible to have either soft or hard processes. Most of the time the hard processes are accompanied by soft interactions, occurring along with the same hadron interaction. While the hard processes such as the Higgs boson production or the high p_T jet production are well understood using perturbative QCD, the soft processes as the underlying event, and the hadronization are not so well understood. In fact, these processes involving a low energy scale at which perturbative series expansion of QCD breaks down, are studied with non-perturbative models. The introduction of the MC simulation is required in order to have a complete description of these two classes of processes and also of their interconnection through the parton showering process.

This chapter, in the first part, focuses on a theoretical introduction to some basic elements of perturbative QCD theory: the QCD factorization theorem, the fixed-order approach in perturbative QCD calculations for the partonic cross-section and the Parton Distribution Functions (PDFs).
In the second part, it is described the parton shower where the parton emission is described in terms of splitting probability for the Initial State Radiation and Final State Radiations. Then, the effects of Multi Parton Interactions and other important processes are described in the simulation of high energy physics events. Lastly, a look at the hadronization process and at the model implemented in PYTHIA is given. The chapter ends with a description of the ideas behind the validation and tuning of MC event generators.

2.1 Structure of Monte Carlo event generators

Monte Carlo event generators build up the structure of a hadron-hadron collision in many different steps. Firstly, a primary hard subprocess is generated and to it is attached the parton shower associated to the incoming and outgoing coloured participants. Then, all the non-perturbative interactions that convert the showers into outgoing hadrons and connect them to the incoming beam hadrons, also the extra interactions that give rise to the UE are simulated. Lastly, all the processes at the hadron level take part in the simulation, e.g. the decays of unstable particles. A schematic representation of all these processes is shown in Fig. 2.1.

Usually, one is interested in the simulation of events of a particular type. The strategy followed is to select the type of hard process and generate the event on

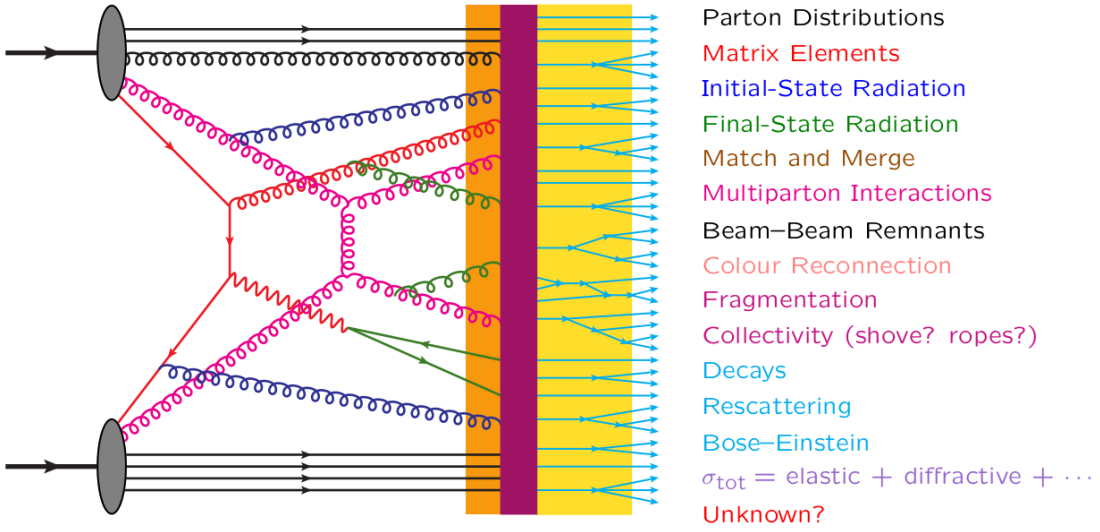


Figure 2.1: A schematic representation for a pp collision. Reading the image from left to right one can have an idea of the evolution of the system. The two incoming hadrons enter the scattering from the left side, the red line indicates the main hard scattering and the magenta one the second parton scattering (MPI) each interaction is associated with initial (blue) and final (green) state radiation, the unscattered partons (black lines) re-enter the color reconnection and hadronization processes. Then the new-formed hadrons (light blue) can undergo to different decays.

the basis of this selection, rather than simulating all types of events and selecting only the one of interest, in fact, this would lead to a very low statistics in the case of rare events. Due to the fact that partons entering in the hard scattering are coloured, they can radiate gluons, this gluon can radiate other gluons or produce quark-antiquark couples, and so on generating showers of outgoing particles. This is simulated by a Markov chain used to choose if add one more parton to the final-state at a certain time. This is called the parton shower algorithm and as will be discussed below it is formulated as a downward evolution in some momentum-transfer-like variable starting from a hard scale defined by the main hard interaction, and as both a forwards evolution of the outgoing partons and a backwards evolution of the incoming partons progressively towards the incoming hadrons.
To be notice that, incoming hadrons are composite particles, i.e. bounded state of partons. So, it is likely to have more than one interaction between partons in the same hadron-hadron collision. Also, these interactions are simulated according to the momentum-transfer-like variable downward evolution.
As the downward evolution goes on is reached the region of order 1 GeV ($\approx \Lambda_{\mathrm{QCD}}$), in which QCD became strongly interacting and perturbation theory breaks down. At this scale, the perturbative description in terms of partons must be terminated and replaced by a non-perturbative hadronization model in order to describe the colour confinement. Unluckily, these models are not derived directly from QCD and so have more free parameters than the preceding ones. But, to a good approximation, they are universal so once they have been tuned to a data set they can be used to predict new collision data.

2.2 QCD factorization theorem

A large number of interesting processes at LHC involve a high transferred-momentum scale, e.g. the production of heavy particles or jets with high transverse momentum. The fundamental idea for the description of hard processes in a hadron-hadron collision is that hadrons are made-up of partons that, at some high energy scale, can be seen as free. So the hadron-hadron interaction can be seen as an interaction between two free partons if a sufficiently high energy scale for the interaction is given.
This idea was developed in the framework of the deep inelastic scattering, and the Bjorken scaling observation confirms it [9]. The generalization of this concept leads to the *QCD factorization theorem.*
The factorization theorem was introduced first by Drell and Yan [10]. The hadron-hadron scattering cross-section is described in terms of partons extending the formalism used for deep inelastic scattering, through the following factorized formula:

$$\sigma_{AB} = \int dx_a \, dx_b \, f_{a/A}(x_a, Q^2) \, f_{b/B}(x_b, Q^2) \, \hat{\sigma}_{ab\to X} \quad . \tag{2.1}$$

Where X is a partonic/leptonic state and $a\,(b)$ a quark, an antiquark or a gluon in the hadron $A\,(B)$, e.g. the proton. The σ_{AB} is the hadronic cross-section while $\sigma_{ab\to X}$ is the partonic cross-section that can be calculated using perturbative QCD. Finally, the $f_{a/A}\,(f_{b/B})$ is the parton distribution function in the hadron $A\,(B)$ for the parton of type a. PDF is dependent on $x_a\,(x_b)$ that is the light-cone momentum fraction carried by the parton with respect to the total momentum of the parent hadron $A\,(B)$; and on the scale of the interaction Q^2.
All the contributions to this formula are described in more detail one by one in the following sections.

2.2.1 Parton distribution functions

The first ingredient in the recipe is the knowledge of the quark and gluon distributions inside the two hadrons that undergo the scattering.
The PDFs used to describe a hard scattering are solutions to the DGLAP (Dokshitzer–Gribov–Lipatov–Altarelli–Parisi) equations [11, 12, 13, 14]

$$\mu_F^2 \frac{\partial f_{i/H}(x,\mu_F^2)}{\partial \mu_F^2} = \sum_j \frac{\alpha_s(\mu_F^2)}{2\pi} \int_x^1 \frac{dz}{z} \, P_{i\to j}(z) \; f_{j/H}\left(\frac{x}{z}, \mu_F^2\right) \quad , \tag{2.2}$$

where μ_F is the energy scale known as *factorization scale*, and $f_{i/H}(x,\mu_F)$ the PDF for the parton of type i inside the hadron H. The PDF is a function of the momentum fraction x and of the energy scale. While, $P_{i\to j}$ are the so-called *splitting functions*: they are the probability that a parton of type i emits a parton j (a quark or a gluon) carrying a fraction z of the i-parton momentum.
The splitting functions have the following perturbative expansions in α_s:

$$P_{i\to j}(x,\alpha_s) = P^{(0)}_{i\to j}(x) + \frac{\alpha_s}{2\pi} P^{(1)}_{i\to j}(x) + \ldots \quad . \tag{2.3}$$

The information on the quark distribution inside a hadron $f_{q/p}(x,Q^2)$ arises from lepton-hadron DIS experiments and lepton-pair production in hadron-hadron

collisions (Drell-Yan processes), while to study gluon distribution, $f_{g/p}(x, Q^2)$, information arising from jet measurements are used. All these quantities are the experimental input in order to evaluate the PDF inside the hadron while the Q-evolution is described by the DGLAP equation as discussed above.

A lot of processes are available for the PDFs evaluation and a lot of PDFs set have been generated, as an example Fig. 2.2 shows the NNPDF3.1 set [15] at NNLO for a virtuality $Q^2 = 10$ GeV^2 (left) and $Q^2 = 10^4$ GeV^2 (right). Note that the gluon

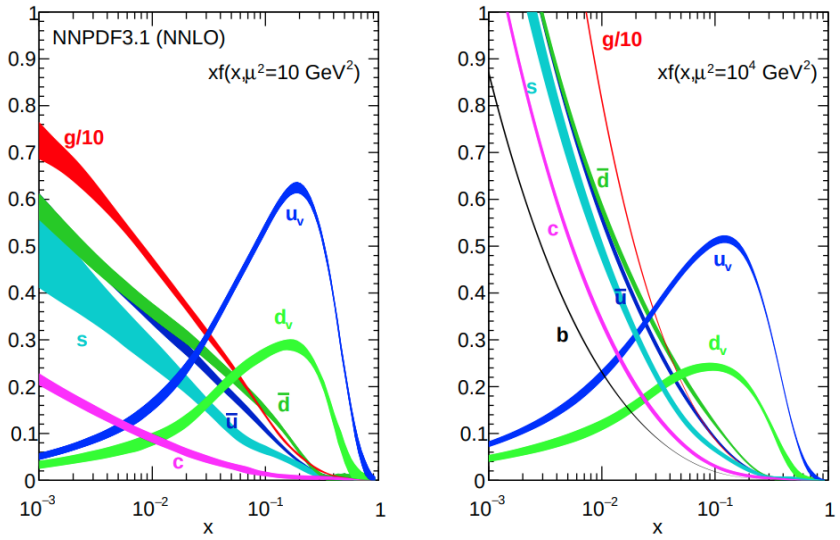


Figure 2.2: The NNPDF3.1 NNLO PDFs set, evaluated at $Q^2 = 10$ GeV^2 (left) and $Q^2 = 10^4$ GeV^2 (right). At low x the contribution from the gluons is the dominating one, while at higher x the dominant contribution is from the valence quarks. The PDF Q evolution shows that when the proton is probed to higher Q^2 the resolution increases (higher Q correspond to smaller distance resolution) and so one has a bigger contribution from the sea quarks at low x values.

contribution has been scaled by a factor of 10: in fact, in the low x region, $x < 0.01$, the gluon contribution is the dominating one, while at high x value the valence quarks dominate the PDF.
In Fig. 2.2 is also displayed that with increasing virtuality (Q^2) at low x the density of the sea quarks increases: this is related to the fact that the hadrons are probed at higher energy and the probe resolution is proportional to the energy.

$$\text{Resolution} \sim \frac{\hbar}{Q} \quad . \tag{2.4}$$

So, when probed at higher energy, the hadrons appear denser[1] than when probed at lower energy. This, as will be discussed in the below sections, is related to the energy dependence in the number of interactions between partons into a single hadron-hadron collision.

[1]This density growth is not endless but at some point a saturation is reached it is called *parton saturation*. The parton saturation finds an explanation in the observation that at some high energy scale the parton mergers have to eventually compensate for splittings.

2.2.2 Partonic cross-section

Partonic cross-section is another fundamental ingredient in the recipe for the description of hadron-hadron interactions. It can be calculated as a perturbative expansion in α_s from QCD first principles using quantum field theory. So the hadron-hadron cross-section can be written as a sum of different contributions, as shown in the following formula:

$$\sigma_{AB} = \int dx_a\, dx_b\, f_{a/A}(x_a, \mu_F^2)\, f_{b/B}(x_b, \mu_F^2)\, \left[\hat{\sigma}_0 + \alpha_s(\mu_R^2)\hat{\sigma}_1 + \dots\right] \quad . \tag{2.5}$$

Where the dependencies on the two unphysical scales, μ_F and μ_R, have been introduced. These two scales are:

- The *factorization scale* μ_F: this scale is related to the resolution with which the hadron is being probed, and separates long- and short-distance physics processes.
- The *renormalization scale* μ_R: the scale at which is evaluated the strong coupling constant α_s. The dependence of α_s on the renormalization scale is related to different effects such as the vacuum polarization, the quark self-energy, the vertex corrections, and the gluon loop corrections to the elementary three-gluon and four-gluon couplings.

The calculation of such expansion at the leading order (LO) is performed by evaluating all the possible tree-level Feynman diagrams for every process. Then the calculation proceeds by computing the squared matrix element and by integrating it over the available phase space (analytically or numerically).
At this point, some divergences may have already been encountered but they can be avoided by imposing restrictions on the phase space.

Higher order calculations

The LO calculation can describe broad features of a particular process and provide a first estimation of its cross-section; anyway, in many cases this is insufficient.
The main source of uncertainty derives from the LO dependence on the unphysical renormalization and factorization scales. Some processes may contribute only when going beyond the first approximation, and some divergence can be resummed.
The next-to-leading order (NLO) calculation requires all the Feynman diagrams that take an extra α_s. This contribution can arise in two different ways:

- **Virtual**: internal lines in the Feynman diagram, see Fig. 2.3a, that don't originate a real particle in the initial or in the final state (loops);
- **Real**: external lines int the Feynman diagram, see Fig. 2.3b, here the particle is real and participates in the observed initial- or final-state (real particles).

Virtual corrections contain infrared divergences, arising by integrating on the loop circulating momentum, that cancel against infrared singularities given by collinear or soft emissions [16, 17, 18].
A common strategy for the renormalization is dimensional regularization: it consists

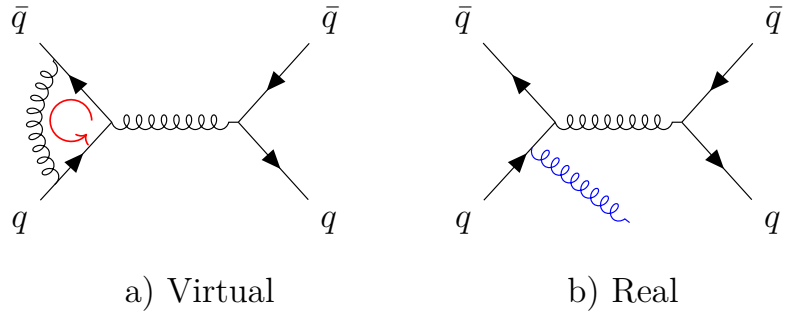


Figure 2.3: Feynman diagrams for a virtual correction (a) the momentum circulating in the loop when integrated on the phase space take to a divergence; and a real correction (b) the particle emitted is real, it participates to the initial or final state of the process.

of performing the calculation in a $D = 4 - 2\epsilon$ dimensional space ($\epsilon < 0$); in that way the singularities appear as single and double poles in ϵ. Then, the limit $\epsilon \to 0$ is taken after the divergences have been cancelled.
This NLO calculation with regularization allows for extending the treatment of these integrals up to zero transverse momentum.

The importance of higher-order calculation is that it allows to describe more processes, as can be shown with the following example. In a Z boson production calculated at:

1) **LO**: the Z is produced without transverse momentum (p_T), and anything can recoil against the Z for momentum conservation (Fig. 2.4a).

2) **NLO**: the Z acquires a finite p_T. In this case the Z boson p_T is balanced by a single parton/gluon (Fig. 2.4b).

3) **NNLO**: the Z p_T can be balanced by two jets (Fig. 2.4c).

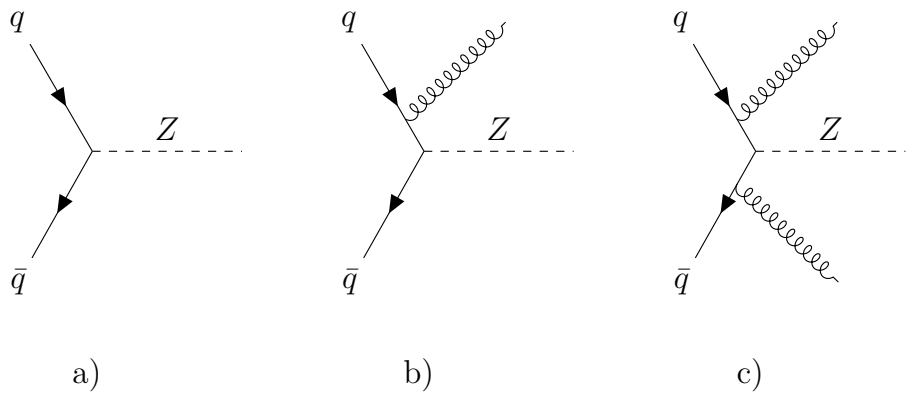


Figure 2.4: Feynman diagrams for the Z production by the annihilation of a quark and an antiquark at LO (a), NLO (b) and NNLO (c). At LO the Z can only be produced with a $p_T = 0$ for the conservation of the momentum.

Another important benefit of performing an NLO calculation is the reduction of the dependence of calculation on the unphysical renormalization (μ_R) and factorization

(μ_F) scales. It is proven that higher-order calculations of observables calculated to order α_s^n are dependent upon the unphysical scales only at order higher than α_s^{n+1} [19]. The range of predictions corresponding to different scale choices is usually attributed to *theoretical uncertainties*, as is shown in Fig. 2.5, where the uncertainties reduce from the LO calculation to the NLO and even more to the NNLO.

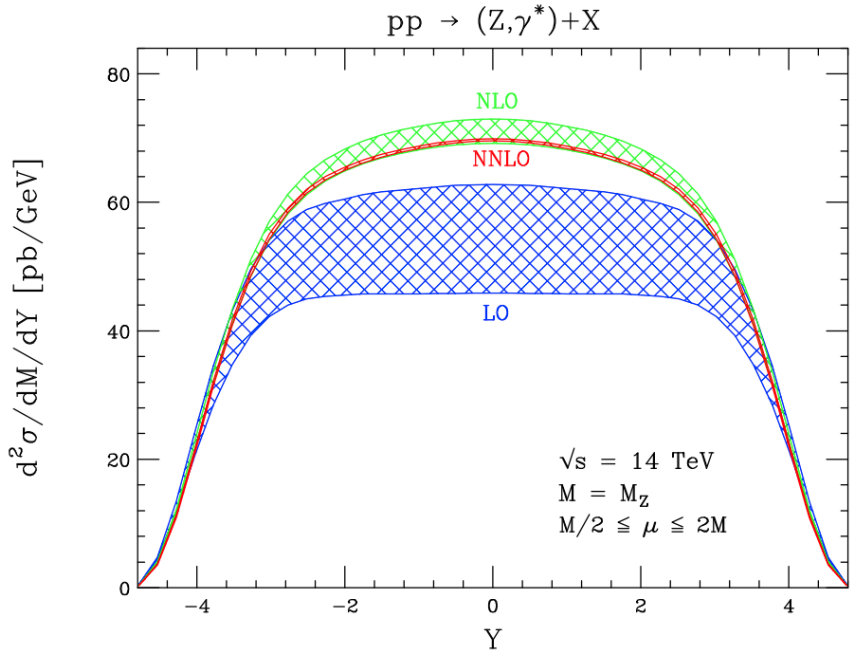


Figure 2.5: The rapidity distribution predictions at LO (blue) NLO (green) and NNLO (red) for the Z production at the center of mass energy $\sqrt{s} = 14$ TeV. The band width is related to the uncertainties. Going from LO to NLO there is an increase in the cross-section prediction and a reduction in the scales uncertainties, the NNLO prediction is in the NLO error band width but there is a further increase in the precision of the prediction. Figure from [19] Section 6.

2.3 All-order approaches and PYTHIA Monte Carlo generator

A different approach to describe the phenomena observed at high-energy colliders, instead of calculating cross-sections order by order in the perturbative expansion, is the use of an *all-order* approach.

The most violent proton-proton collision at the LHC can have a high number of separated jets, $\sim 5 \div 10$. If these jets are observed from a nearer point of view what one can observe is a fractal composition of jets-inside-jets-inside-jets. This structure is expected to continue down to the hadronization scale, a bit below 1 GeV. This fractal nature can lead to hundred of partons in the final-state that then are masked inside the hadrons. The main problem in the theory is that there is no way to perform matrix element calculation to describe such a complicated event topology. Instead, the standard procedure is to use an *all-order* approach: *parton showers*.

2.3.1 Parton Showers

The parton shower algorithm starts from a few partons arising from a hard interaction evaluated by means of the matrix element calculation. Then, as the energy scale at which the scattering is examined decreases these harder partons can split by emission of gluons or quark anti-quark pairs, and so more partons are produced in the shower that can split in their turn. In this process, the higher scale partons are related to the lower hadronization energy scale (close to 1 GeV) using the DGLAP evolution equation formalism. The solution to this equation can be written using a *Sudakov form factor* arising from the probability of no gluon emission in the evolution from higher to lower scale and it ensures the unity for the branching total probability.
In the parton showering process, in addition to the kinematic variables (momentum fraction z and azimuthal angle ϕ) and flavours of the partons, an evolution variable t is generated. PYTHIA8 uses as evolution variable the squared of the relative transverse momentum of the two partons in the splitting (p_T^2). Different choices are made in HERWIG and SHERPA.
As mentioned before, the shower evolution is based on the standard (LO) DGLAP splitting kernels P(z) described here:

$$P_{q\to qg}(z) = C_F \frac{1+z^2}{1-z} \quad ; \tag{2.6}$$

$$P_{g\to gg}(z) = C_A \frac{(1-z(1-z))^2}{z(1-z)} \quad ; \tag{2.7}$$

$$P_{q\to q\bar{q}}(z) = T_R(z^2+(1-z)^2) \quad ; \tag{2.8}$$

where $C_F = \frac{4}{3}$ is the Casimir operator for $SU(3)$, $C_A = N_C = 3$, that are named "color factors", and $T_R = \frac{1}{2}$ that is given by the trace calculation of the group generators, each contribution is multiplied by N_f if summing over all contributing quark flavours.
The parton shower consists of two components: the initial-state radiation describing the emission from the incoming partons and the final-state radiation describing the emission of outgoing partons they are also respectively referred to as spacelike and timelike showers. Both ISR and FSR algorithms are based on these splitting kernels introduced in Eq. 2.6, 2.7 and 2.8. The respective probabilities of emitting radiation as one move in the decreasing evolution variable sequence are:

$$FSR: \qquad \frac{d\mathcal{P}_{FSR}}{dp_T^2} = \frac{1}{p_T^2}\int \frac{dz}{z}\frac{\alpha_s}{2\pi}P(z) \quad ; \tag{2.9}$$

$$ISR: \qquad \frac{d\mathcal{P}_{ISR}}{dp_T^2} = \frac{1}{p_T^2}\int \frac{dz}{z}\frac{\alpha_s}{2\pi}P(z)\frac{f'(x/z,p_T^2)}{f(x,p_T^2)} \quad . \tag{2.10}$$

Where the FSR one represents the infinitesimal probability that a parton branches, or in this case decays, in a dp_T^2 step. Instead, Eq. 2.10 refers to the same probability but it includes the description of the PDF evolution in fact the ISR is described in terms of production of a parton (instead of decays probability as in FSR). The slitting kernels are the same described above except that the splitting probability for the production of a gluon is equal to $2P_{g\to gg}(z)$ since two gluons are generated from the decay of one gluon.

Staring from these infinitesimal probabilities by multiplication of no-emission probability $1 - dP_{ISR}$, or equally $1 - dP_{FSR}$. The Sudakov form factor can be written as:

$$\Delta(p_T^2) = \exp\left(-\int_{p_{T0}^{PS}}^{p'_T} \frac{d\mathcal{P}_{PS}}{dp_T^2}\, dp_T\right) \qquad \text{with} \quad PS = ISR,\ FSR \quad . \tag{2.11}$$

The Sudakov form factor gives the probability of a parton to evolve from an harder scale to a softer scale without emitting a parton harder than some resolution scale. The introduction of the Sudakov form factor resumes all the effects from the soft and collinear gluon emission. For more details and some plots of different Sudakov form factors values see section 3.5 of [19].
The Sudakov factor allows to write the parton shower algorithm using a Markov Chain Monte Carlo method. In fact the system status at a certain value of p_T depends only on the previous status of the system at a $p'_T > p_T$[2].
In PYTHIA8 the contributions from ISR and FSR are interleaved into a single common sequence of decreasing p_T.
It has been seen that the solution to the DGLAP equation is given by putting Eq. 2.9 and Eq. 2.10 in Eq. 2.11 respectively for the ISR and the FSR by a Sudakov form factor that is related to the no-emission probability in the p_T-evolution.
The evolution variables for ISR and FSR are defined starting from the virtuality (Q^2) of the emission:

$$p_T^2 = \begin{cases} (1-z)Q^2 & ISR \\ z(1-z)Q^2 & FSR \end{cases} , \tag{2.12}$$

so, in the two cases: for the FSR $Q^2_{FSR} = (p^2 - m_0^2)$ a time-like virtuality ($p^2 > 0$) is implicated corresponding to a forwards evolution, while for the ISR one has $Q^2_{ISR} = (-p^2 + m_0^2)$ with a space-like virtuality ($p^2 < 0$) describing the backwards evolution. In both cases Q^2 values are positively defined.
The two cutoffs on the transverse momentum scale for ISR, p_{T0}^{ISR}, and FSR, p_{T0}^{FSR}, are free parameters in PYHTIA and are called `SpaceShower:pT0Ref` and `TimeShower:pT0min`.

The parton showers alone are certainly not the whole story in describing the exclusive structure of an event. They are based on collinear and soft approximation. A good description of the hard wide-angle emission and multi-jet final-states cannot be ignored in order to describe many of the observables of interest. So, it is important to perform an higher-order matrix element calculations to describe better all these observables.

2.3.2 Merging parton showers and NLO matrix element calculations

Regions dominated by soft and collinear gluon emissions are described very well by parton showers approach; on the other hand, regions, where partons are energetic

[2]It is important to remember that the simulation is described in a downward evolution in the evolution variable, p_T. So the previous state of the system is located at a higher value of this variable.

and widely separated, are well described by matrix element calculations and not by parton showers. So, the best approach would be to combine the two different descriptions, and this is what is performed in the Monte Carlo generators. But to combine NLO matrix element calculation with parton showers the naive procedure of simply adding a parton shower to an event generated with a matrix element generator does not work.
One problem arises from the fact that tree-level matrix elements are *inclusive*, i.e. they give the probability of having at least n partons in a state calculated exactly to the lowest order in α_s, while the corresponding state generated by a parton shower is *exclusive*, i.e. gives the probability that there are exactly n partons calculated approximately to all orders in α_s
The second problem is to avoid the *double-counting* of some regions of phase space or, conversely, the undercount other regions.
To solve these problems there are different strategies. The various strategies are divided into two groups. The ones referred to as *"match"* are those approaches in which high-order corrections to an inclusive process are integrated with the parton shower. The other strategy involves a *"merge"* this is done by defining a merging scale, where any parton produced above that scale is generated with a corresponding higher-order matrix element and any parton produced below is generated by the shower.
An example of merging approach is the FxFx merging scheme, developed by Frixione, Nason, Webber in the MC@NLO framework [20, 21, 22, 23]. In this scenario the risk of double counting is given by the fact that at NLO one emission can be made explicit as indicated in Fig. 2.6 by the red gluon line, and then the progress of the parton shower can leads to a double counting between real emission matrix element and the parton shower as shown in Fig. 2.6, the double-counting sources are indicated by the blue arrows.

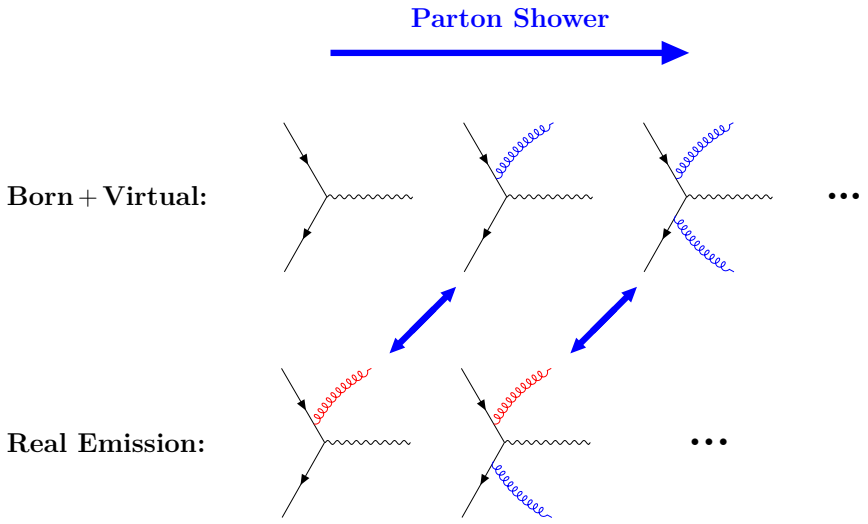


Figure 2.6: The FxFx margin scheme have to avoid this double counting. Feynman diagrams that can lead to a double counting are grouped with a blue arrow, the blue emission are related to the parton shower while the red ones to the NLO process.

So, it is clear that the matrix element calculations cannot be blindly combined with parton showers. The combination of these two approaches is a very active research topic, and is important for giving a good description for jet production from QCD.

2.4 Soft QCD processes

Lots of physics processes and modeling issues are related to as "soft QCD". In the following, firstly, is performed a brief introduction of different QCD processes that form the dominant part of the total hadron-hadron cross-section. Then, the model of the Multi Parton Interaction and its implementation in pyhtia8 are described. Lastly, the primordial k_T concept, the color reconnection and the hadronization processes are introduced. All these contributions have to be understood if one wants to have a good description of the soft QCD contribution to what is observed, in particular for the description of the UE.

2.4.1 Various contributions to hadron-hadron cross-section

Hadron-hadron scatterings are classified by the characteristics of the final states. The main division is between *elastic* and *inelastic* scatterings. In elastic ones:

$$A(p_A) + B(p_B) \longrightarrow A(p'_A) + B(p'_B) \quad , \tag{2.13}$$

both the hadrons A and B emerge intact and no other particles are produced. The only exchanged quantity is momentum. Fig. 2.7 shows how the topology of an observed elastic event would be.

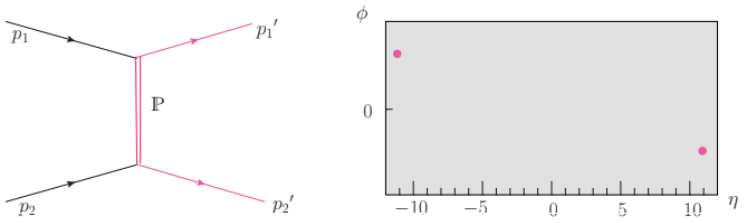


Figure 2.7: Elastic scattering diagram and ϕ-η plot showing the angular distribution of products after the interaction. Figure from [24]

All the other cases are classified as inelastic. In inelastic scatterings the final-state is different from the initial one.

$$A(p_A) + B(p_B) \longrightarrow X \quad . \tag{2.14}$$

Where X indicates that a new final state has been generated in the interaction. In this case, the system gets complex and new particles are produced.
So, the total cross-section can be written-out as a sum of two contributions:

$$\sigma_{\text{tot}}(s) = \sigma_{\text{el}}(s) + \sigma_{\text{inel}}(s) \quad . \tag{2.15}$$

Where the various contribution depend on the center-of-mass energy squared $s = (p_A + p_B)^2$.

The inelastic contribution can be further subdivide into *diffractive* and *non-diffractive* events. The division is based on the presence, or not, of a large rapidity gap somewhere in the final-state that can be seen as a decay of an excitation of the beam particles.

2.4.2 Diffractive cross-section

Once an event has been classified as diffractive it is possible to distinguish between three possible classes of diffractive events: *double-diffractive* (DD), *single-diffractive* (SD) and *central-diffractive* (CD). In DD events both the particles of the beams are excited and both are fragmented in the collision (Fig. 2.8b), on the other hand in SD dissociation just one of the two hadrons get excited while the other survives the collision intact (Fig. 2.8a). In the last case, CD, both the beam particles survive intact, but they leave an excited state in the central region that then decays (Fig. 2.8c).
So, the inelastic cross-section in Eq. 2.15 can be rewritten as:

$$\sigma_{\text{inel}}(s) = \sigma_{\text{SD}}(s) + \sigma_{\text{DD}}(s) + \sigma_{\text{CD}}(s) + \sigma_{\text{ND}}(s) \quad , \tag{2.16}$$

where ND are all the *non-diffractive* inelastic events[3] (Fig. 2.8d), these types of events are the dominants in proton-proton collisions at LHC.

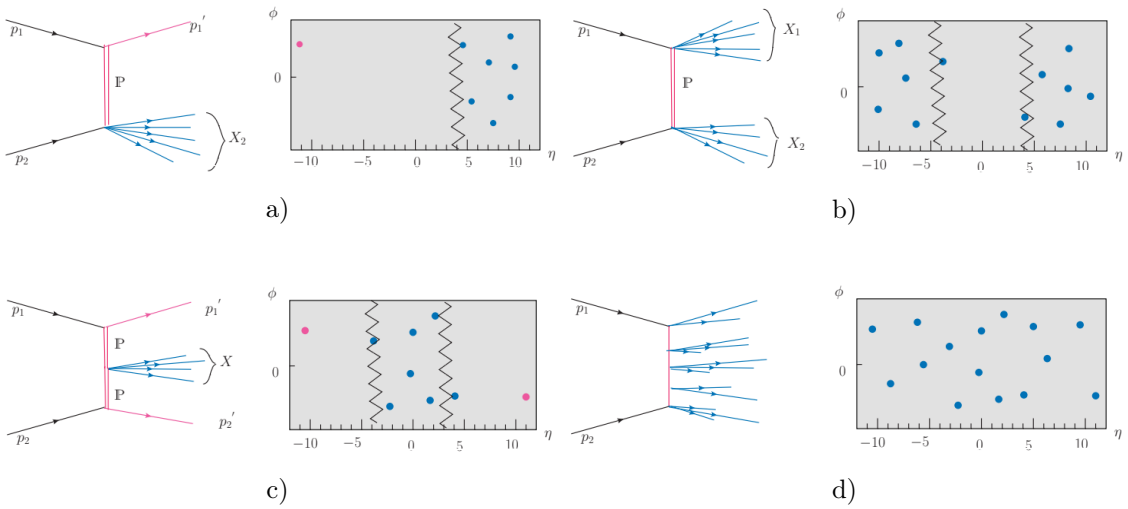


Figure 2.8: The figure shows: a) an SD event with a rapidity gap in the region $-10 < \eta < 3.5$; b) a DD event where all the two hadrons fragment and is formed a rapidity gap $|\eta| < 3.5$; c) the two hadrons survive but a central excitation is produced (CD event); d) an ND event all the $\phi - \eta$ region is covered by the products of the collision. Figure from [24]

The next section will focus on the description of the non-diffractive part of the total cross-section using the MPI model. In very rough terms, this means that the non-diffractive cross-section is saturated by more than a single partonic scattering

[3]It has to be mentioned that other topologies are possible defined as multi-diffractive events whit various rapidity gap observed. These types of events have a very low cross-section ($\ll 1$ mb) with respect to the other considered.

and that the number of partonic interactions is determined by a Poisson distribution. This model has been very successful in the description of many experimental measurements at hadron colliders.

2.5 Multiple Parton Interactions in PYTHIA

Since hadrons can be viewed as "bunches" of partons, it is likely that in the same hadron-hadron collision more than one pair of partons can undergo to a scattering. This phenomenon is known as *Multiple Parton Interactions* and is related to the composite nature of the incoming hadrons.
So, it is clear that at some level these MPI have to exist, and they can become important in the description of the event; they can change the color topology of the colliding system as a whole.
In this scenario, it is important to have a good understanding of the phenomenon. The aim of this section is to describe the basic concepts that are used to simulate MPI, for example in PYTHIA Monte Carlo event generator; then some focus is given to the free parameters that will be tuned in the following sections.

2.5.1 Basic Concepts

The main hypothesis of the MPI model is the validity of the QCD factorization theorem not only for the hard process but also for the other partons scatters.
So, one can write the following formula for the interaction differential cross-section for the hadron-hadron collisions:

$$\frac{d\sigma_{\text{int}}}{dp_T} = \sum_{i,j,k,l} \int dx_1 \int dx_2 \int d\hat{t}\, f_i(x_1, Q^2) f_j(x_2, Q^2) \frac{d\hat{\sigma}_{ij \to kl}}{d\hat{t}} \delta\left(p_\perp^2 - \frac{\hat{t}\hat{u}}{\hat{s}}\right) \quad , \quad (2.17)$$

where $\frac{d\hat{\sigma}_{ij \to kl}}{d\hat{t}}$ is the differential cross-section for QCD hard $2 \to 2$ processes, this processes are the one reported in Table 2.1.
In Eq. 2.17 the Mandelstam variables associated with the partonic system are used. They are defined as:

$$\hat{s} = (p_1 + p_2)^2 = (p_3 + p_4)^2 = x_1 x_2 s \quad (2.18)$$

$$\hat{t} = (p_1 - p_3)^2 = (p_2 - p_4)^2 \quad (2.19)$$

$$\hat{u} = (p_1 - p_4)^2 = (p_2 - p_3)^2 \quad (2.20)$$

where p_1, p_2 are the four-momenta of the incoming partons and p_3, p_4 the four-momenta of the outgoing partons.
It has been assumed also that the "hardness" of a process is defined by the p_T scale of the interaction ($Q^2 = p_T^2$).
As one can see from the formulae in Table 2.1 at small scattering angles, for $t \to 0$, the t-channel gluon exchange processes $qq' \to qq'$, $qg \to qg$ and $gg \to gg$ dominate the full matrix element. For scatterings that are soft relative to $\hat{s}$, $|\hat{t}| \ll \hat{s}$, it is possible to approximate $|\hat{t}|$ as:

$$p_T^2 = \frac{\hat{t}\hat{u}}{\hat{s}} = \frac{\hat{t}(-\hat{s} - \hat{t})}{\hat{s}} \approx |\hat{t}| \quad , \quad (2.21)$$

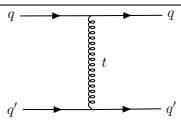
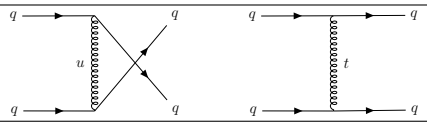
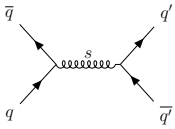
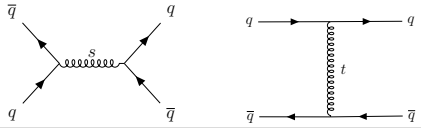
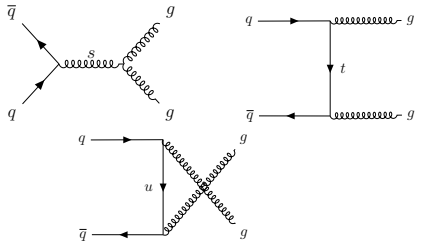
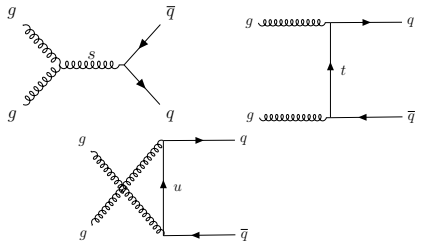
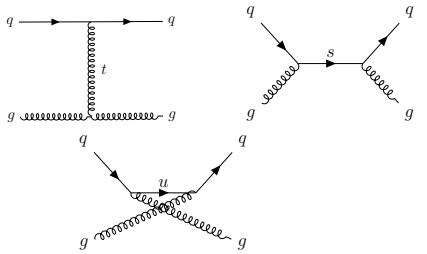
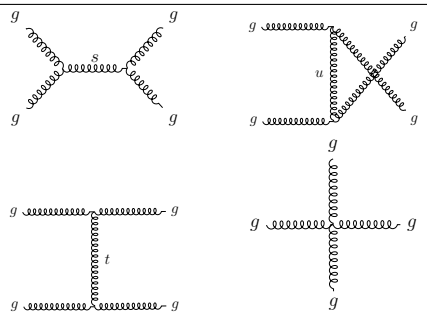
Process	Amplitude	$\sum \lvert\mathcal{M}\rvert^2/(4\pi\alpha_s)^2$
$qq' \rightarrow qq'$		$\frac{4}{9}\frac{s^2+u^2}{t^2}$
$qq \rightarrow qq$		$\frac{4}{9}\frac{s^2+u^2}{t^2} + \frac{4}{9}\frac{s^2+t^2}{u^2} - \frac{8}{27}\frac{s^2}{tu}$
$q\overline{q} \rightarrow q'\overline{q'}$		$\frac{4}{9}\frac{t^2+u^2}{s^2}$
$q\overline{q} \rightarrow q\overline{q}$		$\frac{4}{9}\frac{s^2+u^2}{t^2} + \frac{4}{9}\frac{t^2+u^2}{s^2} - \frac{8}{27}\frac{u^2}{st}$
$q\overline{q} \rightarrow gg$		$\frac{32}{27}\frac{t^2+u^2}{tu} - \frac{8}{3}\frac{t^2+u^2}{s^2}$
$gg \rightarrow q\overline{q}$		$\frac{1}{6}\frac{t^2+u^2}{tu} - \frac{3}{8}\frac{t^2+u^2}{s^2}$
$qg \rightarrow qg$		$-\frac{4}{9}\frac{s^2+u^2}{su} + \frac{s^2+u^2}{t^2}$
$gg \rightarrow gg$		$\frac{9}{2}\left(3 - \frac{tu}{s^2} - \frac{su}{t^2} - \frac{st}{u^2}\right)$

Table 2.1: Parton-Parton differential cross-sections ($2 \rightarrow 2$ QCD process), can beh calculated in pQCD by evaluating the matrix element for each process involving quark, antiquark and gluon. Table from [25]

in this limit, the only differences between quark and gluon cross-sections are the color factors.

$$\hat{\sigma}_{gg} : \hat{\sigma}_{qg} : \hat{\sigma}_{qq} = \frac{9}{4} : 1 : \frac{4}{9} \quad . \tag{2.22}$$

So, the Eq. 2.17 can be rewritten like:

$$\frac{d\sigma_{int}}{dp_T^2} \approx \int\int \frac{dx_1}{x_1}\frac{dx_2}{x_2} F(x_1, p_T^2)\, F(x_2, p_T^2) \frac{d\hat{\sigma}_{2\to 2}}{dp_T^2} \quad , \tag{2.23}$$

whit:

$$\frac{d\hat{\sigma}_{2\to 2}}{dp_T^2} = \frac{8\pi\alpha_s^2(p_T^2)}{9p_T^4} \quad ; \tag{2.24}$$

$$F(x, Q^2) = \sum_q \left(x\, q(x, Q^2) + x\, \overline{q}(x, Q^2) \right) + \frac{9}{4}\, x\, g(x, Q^2) \quad . \tag{2.25}$$

Now, the Eq. 2.23 can be integrated:

$$\sigma_{int}(p_{T\,min}) = \int_{p_{T\,min}^2}^{(\sqrt{s}/2)^2} \frac{d\hat{\sigma}_{2\to 2}}{dp_T^2}\, dp_T^2 \propto \frac{1}{p_{T\,min}^2} \xrightarrow{p_{T\,min}\to 0} \infty \quad . \tag{2.26}$$

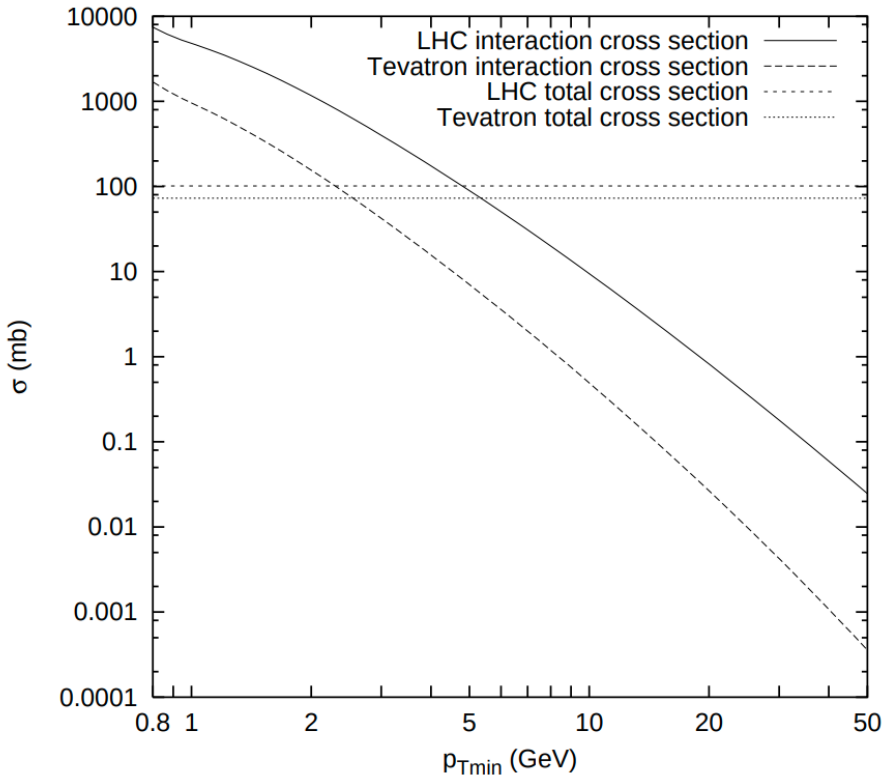


Figure 2.9: This figure shows the interaction cross-section ($\sigma_{\rm int}$) at Tevatron ($p\overline{p}$, $\sqrt{s}$ = 1.8 TeV) and at LHC (pp, $\sqrt{s}$ = 14 TeV). The flat lines are the corresponding values for the total cross-section. The interaction cross-section that arise from Eq. 2.26 is divergent for $p_{T\,min} \to 0$ in reality a dumping of this divergence is expected due to the color screening effect.

The total cross-section is divergent in the limit $p_T \to 0$. Fig. 2.9 shows the rise of the interaction cross-section as the energy scale decreases. Due to this divergence, the total interaction cross-section at some p_T scale can exceed the total proton-proton cross-section (described in the section above).
To understand this paradox should be noted that the interaction cross-section described in Eq. 2.26 is related to the interaction probability between two partons and counts the number of interactions, while the total proton-proton cross-section σ_{pp} counts the number of events. For example, an event (a proton-proton collision) in which two partons interact counts once in the total cross-section and twice in the interaction cross-section.
So, the ratio between these two quantities is perfectly allowed to be larger than unity, it can be written as:

$$\frac{\sigma_{int}(p_{T\,min})}{\sigma_{tot}} = \langle n \rangle (p_{T\,min}) \quad . \tag{2.27}$$

Furthermore, the *screening effect* has to be considered, in fact, the incoming hadrons are color singlet objects but the partons are not. Therefore, when the p_T of an exchanged gluon is small, and so the associated wavelength large, this gluon can no longer resolve the color charges and the effective coupling is decreased, this screening set a cutoff in the divergence. The screening effect is schematically shown in Fig. 2.10.

a) Screened color charges b) Resolved color charges

Figure 2.10: A picture of the screening effect. The left figure shows two color charges that are not resolved by the gluon in fact the wavelength is greater than the spatial separation, d, of the two charges. So the resolution of the probe is not enough to discriminate among the various color charges. While on the right figure the two charges are very well distinct.

So, this cutoff is associated with color screening distance i.e. the average size of the region within which the compensation of color charge occurs. This cutoff is then introduced in the factor as:

$$\frac{\alpha_s^2\,(p_{T0}^2 + p_T^2)}{\alpha_s^2\,(p_T^2)} \frac{p_T^4}{(p_{T0}^2 + p_T^2)^2} \quad . \tag{2.28}$$

This factor contains the phenomenological regularization of the divergence, with the factor p_{T0} that has to be tuned to data.
Now the interaction cross-section is smoothly regularized and therefore finite.
To be notice that: parameter p_{T0} does not have to be energy-independent, but since the energy is related to the sensitivity of the probe, higher energy is related to the

capacity of probing PDFs to lower x values (as discussed previously, see Fig. 2.2) and in this low-x region, the number of partons rapidly increases. So, the partons are more closely packed in this region and as a consequence the color-screening distance decrease.
The number of partons is related to x with a power-law so, it is likely to have a dependence of the same form for p_{T0} with respect to the center-of-mass energy

$$p_{T0}(\sqrt{s}) = p_{T0}^{ref} \left(\frac{\sqrt{s}}{E_{CM}^{ref}} \right)^{E_{CM}^{pow}} \quad . \tag{2.29}$$

These two parameters are named `MultiplepartonInteractions:pT0Ref` and `MultiplepartonInteractions:ecmPow` and control respectively the threshold for the MPI to occur at a fixed center-of-mass energy and the power evolution with $\sqrt{s}$.

In PYTHIA8 MPI, as said before, are also generated in a decreasing p_T sequence. So the hardest MPI is generated first. Then the probability for an interaction, i, to occur at a scale p_T can be written using a Sudakov-type expression (as was done for the ISR and the FSR):

$$\frac{d\mathcal{P}_{MPI}}{dp_T} = \frac{1}{\sigma_{ND}} \frac{d\sigma_{2\to 2}}{dp_T} \exp\left(- \int_{p_T}^{p_T\, i-1} \frac{1}{\sigma_{ND}} \frac{d\sigma_{2\to 2}}{dp'_T}\, dp'_T \right) \quad . \tag{2.30}$$

2.5.2 Momentum and flavour conservation

One problem in MPI simulation is to achieve momentum conservation, so for every consecutive interaction, it has to be taken into account the modification in the PDFs by the preceding, $i-1$, interaction. To do that in PYTHIA the PDF are rescaled to the remaining available x range, adjusting their normalization.
One needs to take into account the momentum fraction x_i removed from the hadron remnant by the $i-th$ interaction. This is done by evaluating PDF not at x_i but at a rescaled value

$$x'_i = \frac{x_i}{X} \qquad \text{with} \quad X = 1 - \sum_{j=1}^{i-1} x_j \quad . \tag{2.31}$$

So, by using these quantities, the PDFs can be rewritten as:

$$f_i(x, Q^2) \;\longrightarrow\; \frac{1}{X} f_0\left(\frac{x}{X}, Q^2\right) \quad , \tag{2.32}$$

where f_0 is the original one-parton inclusive PDFs.
Now, requiring also the flavour conservation and taking into account the number of valence and/or sea quarks involved in the preceding MPI. The full forms of the PDFs used for the $i-th$ MPI are:

$$f_i(x, Q^2) = \frac{N_{fv}}{N_{fv0}} \frac{1}{X} f_{v0}\left(\frac{x}{X}, Q^2\right) + \frac{a}{X} f_{s0}\left(\frac{x}{X}, Q^2\right) + \sum_j \frac{1}{X} f_{c_j 0}\left(\frac{x}{X}, Q^2\right) \quad , \tag{2.33}$$

$$g_i(x, Q^2) = \frac{a}{X} g_0\left(\frac{x}{X}, Q^2\right) \quad , \tag{2.34}$$

where:

- $f_i(x, Q^2)$ ($g_i(x, Q^2)$) are the squeezed PDFs for quarks (gluons);
- N_{fv} the number of remaining valence quarks of the given flavour;
- N_{fv0} the number of original valence quarks of the given flavour (for the proton they are: $N_u = 2$, $N_d = 1$);
- f_{s0} the sea-quark PDF;
- f_{cj} the companion PDF, this arise from the splitting $g \to q\overline{q}$ and a quark j is kicked out.

The factor a is defined to satisfy the total momentum sum rule.

2.5.3 Impact Parameter Dependence

The simplest hypothesis for the simulation of multiple interactions is to assume the same initial state for all hadron collisions without dependencies on the impact parameter.
The more realistic scenario is to include the possibility that each collision could be characterized by a different impact parameter b, where a small b value corresponds to a large overlap between the two hadrons this is related to the probability of parton-parton interaction to take place.
To include the impact parameter dependence on the collision, it is necessary to make some assumptions on the matter distribution inside the proton. A possibility is to assume a spherically symmetric distribution inside an hadron at rest $\rho(\mathbf{x})\, d^3x = \rho(r)\, d^3x$. A Gaussian ansatz is the most simple choice but it appears to lead to a narrow multiplicity distribution and a too little pedestal effect. So the choice is a double Gaussian:

$$\rho(r) \propto \frac{1-\beta}{a_1^3} \exp\left\{-\frac{r^2}{a_1^2}\right\} + \frac{\beta}{a_2^3} \exp\left\{-\frac{r^2}{a_2^2}\right\} \quad , \tag{2.35}$$

where a fraction β of matter is contained in a smaller central region, called *core*, of radius a_2, while the rest of the matter fraction in a larger one of radius a_1. In PYTHIA8 the two parameters related to the fraction of matter and the spatial extension of the core region are named respectively as `MultipartonInteractions:coreFraction` and `MultipartonInteractions:coreRadius`.
Now, for a given matter distribution $\rho(r)$, the time-integrated overlap function of the incoming hadrons during the collision is given by:

$$\mathcal{O}(b) = \int dt \int d^3x\, \rho(x, y, z)\, \rho(x + b, y, z + t) \quad . \tag{2.36}$$

Assuming the matter distribution function in Eq. 2.35 and inserting it into Eq. 2.36, one obtains the following expression:

$$\mathcal{O}(b) \propto \frac{(1-\beta)^2}{2a_1^2} \exp\left\{-\frac{b^2}{2a_1^2}\right\} + \frac{2\beta(1-\beta)}{a_1^2 + a_2^2} \exp\left\{-\frac{b^2}{a_1^2 + a_2^2}\right\} + \frac{\beta^2}{2a_2^2} \exp\left\{-\frac{b^2}{2a_2^2}\right\} \tag{2.37}$$

This quantity is useful to quantify the effect of overlapping protons. The larger is $\mathcal{O}(b)$ the more probable are parton-parton scatters between the incoming protons.

So, it is natural to assume a linear relationship between the overlapping function and the mean number of interactions in the event as in the following equation:

$$\langle \widetilde{n} \rangle = k\mathcal{O}(b) \quad . \tag{2.38}$$

In this scenario, the number of interactions at a fixed impact parameter value is given by a Poissonian distribution:

$$\mathcal{P}(\widetilde{n}) = \langle \widetilde{n} \rangle^{\widetilde{n}} \ \frac{e^{-\langle \widetilde{n} \rangle}}{\widetilde{n}!} \tag{2.39}$$

If the matter distribution has an infinite tail (like the one in Eq. 2.35) events may be obtained with arbitrarily large b values. This can be a problem for the definition of the total hadron-hadron cross-section.
So, imposing that at least one parton interaction occurs in the hadron-hadron collision, it is ensured that a finite total cross-section is obtained. The probability that at least one interaction occurs by a hadron scattering with impact parameter b is:

$$\mathcal{P}_{\text{int}} = \sum_{\widetilde{n}=1}^{\infty} \mathcal{P}_{\widetilde{n}(b)} = 1 - \mathcal{P}_0 = 1 - e^{-\langle \widetilde{n}(b) \rangle} = 1 - e^{-k\mathcal{O}(b)} \tag{2.40}$$

Now, the number of interaction, for impact parameter b, per event is give by:

$$\langle \widetilde{n}(b) \rangle = \frac{\langle k\mathcal{O}(b) \rangle}{1 - e^{-k\mathcal{O}(b)}} = \frac{\langle k\mathcal{O}(b) \rangle}{\mathcal{P}_{\text{int}}(b)} \tag{2.41}$$

where it has been divided for the total interaction probability to take into account for the request of at least one parton interaction. So, this have modified the probability distribution of interactions number from a Poissonian to a narrower one at each b fixed.

2.5.4 Parton rescattering

It is not necessary that the partons undergoing MPI are a different partons couple from the one scattered before. As shown in Fig. 2.11 MPI can also arise when a parton scatters more than once against partons from the other beam, this is called *parton rescattering*.
MPI can take place in three different ways:

1. No one of the partons that enter in the second scattering undergoes to another scatter before (Fig. 2.11 left);

2. Only one of the two partons has already been scattered (Fig. 2.11 middle);

3. Both the partons have already been scattered before (Fig. 2.11 right).

The second and the third are the rescatters. The overall influence of rescatters in proton-proton interactions was estimated to be small with respect to the first case with distinct $2 \to 2$ scatters.

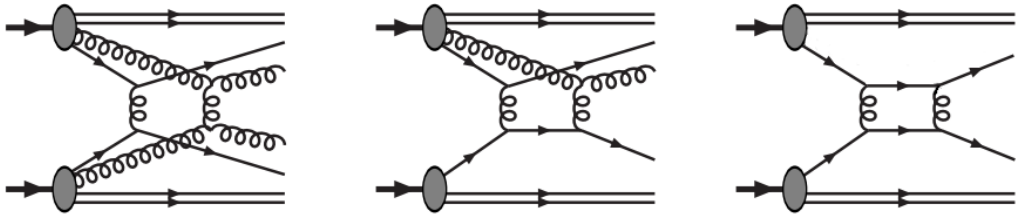


Figure 2.11: This figure shows parton rescattering. On the left image a double $2 \to 2$ scattering (no rescattering) is shown; on the middle a parton rescattering process where only one of the rescattered partons has already scattered; on the right image a parton rescattering with both the partons that undergo to rescattering have already been scattered.

The simulation of partons rescattering starts from the evaluation of the parton density as:

$$f(x,Q^2) \quad \longrightarrow \quad \underbrace{f_{\text{rescaled}}(x,Q^2)}_{\text{hadron remnant}} + \underbrace{\sum_{i=1}^{N} \delta(x-x_i)}_{\text{scattered parton(s)}} \quad , \tag{2.42}$$

where the $\delta(x - x_i)$ takes into account the N already-scattered partons that have a fixed momentum fraction x_i. While the hadron remnants that do not enter a scattering before are still described by a continuous momentum density, but rescaled to achieve momentum conservation once that the already-scattered partons have been extracted:

$$\int_0^1 \left(f_{\text{rescaled}}(x,Q^2) + \sum_n \delta(x-x_n) \right) dx = 1 \quad . \tag{2.43}$$

2.5.5 Interplay of Multiple Interaction and Parton Shower

Obviously, the parton shower can be applied not only to the partons that undergo to the primary hard scattering, but also the other partons especially the ones interested by the MPI.
In other words, the event structure of a hadron-hadron collision can be very complex, e.g. a single initial-state parton can split into two before both of them enter a hard collision. At the same time, an independently resolved parton may undergo another collision, while all of them collectively radiate further gluons. This is a very complex situation that cannot be described exactly. But a good description can be obtained by considering an *interleaved* parton shower and MPI evolution. An example of a possible resulting event structure is reported in Fig. 2.12.
The total interaction probability is given from the composition of the various contributions from each process:

$$\begin{aligned} \frac{d\mathcal{P}}{dp_T} = &\left(\frac{d\mathcal{P}_{MPI}}{dp_T} + \sum \frac{d\mathcal{P}_{ISR}}{dp_T} + \sum \frac{d\mathcal{P}_{FSR}}{dp_T} \right) \times \\ &\times \exp\left\{ - \int_{p_T}^{p_T\, i-1} \left(\frac{d\mathcal{P}_{MPI}}{dp'_T} + \sum \frac{d\mathcal{P}_{ISR}}{dp'_T} + \sum \frac{d\mathcal{P}_{FSR}}{dp'_T} \right) dp'_T \right\} \end{aligned} \tag{2.44}$$

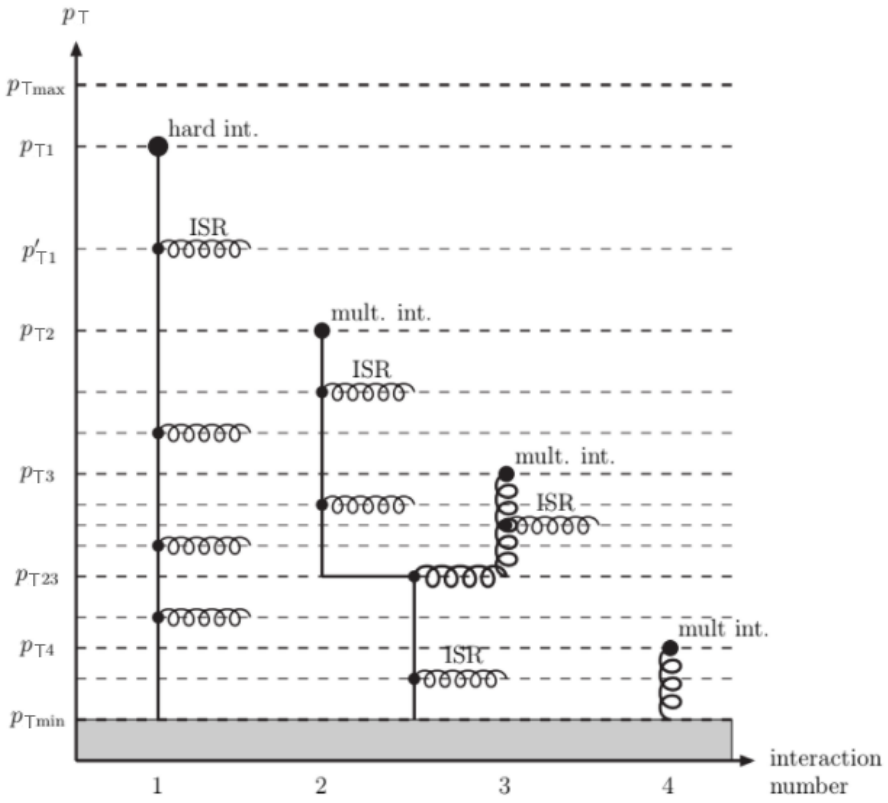


Figure 2.12: An example of Multiple Partons Interactions and associated radiations. The downward evolution in p_T is shown here by reading the graph from top to bottom. The 4 parton-parton interactions occur to p_{T1}, p_{T2}, p_{T3} and p_{T4}.

In Fig. 2.12 are shown 4 parton-parton interactions with their associated showers (ISR and FSR). The downward evolution correspond to read the graph from top to bottom. The first hard interaction occurs at a scale $p_T = p_{T1}$ while the following ones at lower scales p_{T2}, p_{T3}, p_{T4}. Each interaction is associated with is radiation the first one occurs at $p_T = p'_{T1}$. The scatterings at p_{T2} and p_{T3} are originating from the same mother parton.
This diagram is related to one of the two hadrons. The full event can be illustrated if a similar diagram is drawn for the other hadron and connected to the full black circles.

2.6 Primordial k_T and Color reconnection in PYTHIA8

What is left after that the p_T evolution is ended? The evolution in p_T can create an arbitrary complicated final state. This final state contains contributions from the scattered and unscattered partons that don't enter the p_T evolution. The last ones are the so-called *Beam Beam Remnants*. BBR take into account the number of valence quarks remaining and the number of sea quarks required for the overall flavour conservation and to ensure momentum conservation. So, once the evolution

ended, a set of partons is added to the BBR. The BBR have to take all the remaining longitudinal momentum that is not extracted by the MPI initiators to ensure the overall momentum conservation.
The showering process plus the MPI introduction leads to a large number of partons in the final states. Then, in the simulation, all the partons take the contribution from the primordial k_T (originating from the Fermi motion) this is described in the next section.

Primordial k_T

It has been considered only the longitudinal momentum. In a real scenario partons, inside the hadrons, are fermions. So, are expected to have a non-zero initial transverse momentum arising from the Fermi motion inside the incoming hadrons. This is denoted as *primordial* k_T. This is different from the transverse momentum derived from DGLAP shower evolution or from the hard interaction.

Based on Fermi motion alone, one would expect a value of the primordial k_T of the order of the inverse hadron radius:

$$k_T \simeq \frac{\hbar}{r_p} \approx \frac{0.2 \text{ GeV} \cdot \text{fm}}{0.7 \text{ fm}} \approx 0.3 \text{ GeV} \quad , \tag{2.45}$$

but, as an example, to reproduce the data for the p_T distributions of Z bosons produced in hadron-hadron collisions and observed in Drell Yan processes, one needs a larger contribution from the primordial k_T. This phenomenon has not a satisfactory phenomenological explanation. Until a convincing explanation is found the idea is to consider an effective primordial k_T for the initiators larger than the one in Eq. 2.45. This larger value can be seen as a value re-summing also all the low-p_T "unresolved" effects that are not taken into account by the showering and all the above processes.

In PYTHIA the primordial kT is assigned to all shower initiators sampling a Gaussian distributions for p_x and p_y independently with variable width σ

$$\sigma(Q, \widehat{m}) = \frac{Q_{1/2}\,\sigma_{\text{soft}} + Q\,\sigma_{\text{soft}}}{Q_{1/2} + Q} \frac{\widehat{m}}{\widehat{m}_{1/2} + \widehat{m}} \tag{2.46}$$

Where Q is the hard-process renormalization scale for the main hard process and the p_T scale for subsequent MPI. σ_{soft}, σ_{hard} are the minimum and maximum width, $\widehat{m}$ is the invariant mass, while $Q_{1/2}$ and $\widehat{m}_{1/2}$ the halfway values between the two extremes.
In PYTHIA the maximum width parameters is called `BeamRemnants:primordialKThard`.
Note that in PYTHIA8 simulation the addition of the primordial kT to all shower initiators does not automatically guarantee overall p_T conservation. To minimize the mismatch lot of tricks are used in the generator, and as the final step, a brute force shift is imposed on all the final state particles.

2.6.1 Color Reconnection

The last important step at parton level is the *color reconnection*. CR is motivated by the fact that MPI leads to different color strings. In the previous steps, the

planar limit of the QCD was assumed where $N_c \to \infty$. Now, moving to real case $N_c = 3$, during the development of the parton shower, partons are represented and also connected by colour lines as it is shown in Fig. 2.13. Quark and anti-quarks are described by a single color line with an arrow that describes the direction of the color flow, while gluons are represented by a pair of color lines pointing in opposite directions.

Figure 2.13: Color flow in quark-gluon vertices. The Feynman diagrams are shown in black and associated with the respective color connection lines. This figure is taken from [26].

All these color strings that can be overlapped in physical space can be reconnected. The basic idea is to reconnect strings in order to reduce the total string length, and thus the potential energy (Lund Model).
In PYTHIA8 different models for the CR exist. In the default model (*MPI-based model*) the partons are classified according to the MPI system to which they belong[4], the reconnection is performed giving to each system, whit a hardness scale p_T, a probability to reconnect defined by:

$$\mathcal{P}_{\text{rec}} = \frac{p^2_{T\,\text{rec}}}{(p^2_{T\,\text{rec}} + p^2_T)} \qquad p_{T\,\text{rec}} = R \times p_{T\,0} \quad , \tag{2.47}$$

the `ColorReconnection:range`, R, is a user-tunable parameter while $p_{T\,0}$ is the same parameter defined in MPI simulation.
With this definition for the probability, it is clear that systems with low p_T are more likely to reconnect to others, while the higher-p_T systems tend to escape from the interaction point without being reconnected. This idea finds its justification in the fact that lower p_T values correspond to larger spatial extension and so these strings have more chance to overlap with others and so the systems to reconnect.

2.7 Hadronization

The hadronization process takes all these partons (colored strings) and transforms them into a set of experimental-observed color-neutral hadrons.
Hadronization is one of the most non-understood process in high energy physics. Due to the intrinsically non-perturbative nature of the phenomenon, it cannot be derived by QCD fundamental principles. A good description of the hadronization is needed it participate in all the interaction that contain a final-state composed of colored partons. So, necessarily, the simulated final-state depends strongly on the hadronization model used.
There are two main hadronization model classes in current use: the *string* and the *cluster model*. PYTHIA8 is based on the former. The following will describe the string model in more detail.

[4]Note that each parton interaction is originally a $2 \to 2$ scattering. Further details on various models can be found at [26].

2.7.1 String model

In Pythia hadronization is based solely on the *Lund string fragmentation model* [27, 28].
The Lund model's basic idea is to break the color strings that connect the quarks to reduce the total string length, where the string is representative of the potential:

$$V(r) = -\frac{a}{r} + \kappa r \qquad \text{with} \quad \kappa \approx 1 \text{ GeV/fm} \quad , \tag{2.48}$$

where κ is the string tension. This potential is a combination of an attractive (Coulomb) potential and of a linear potential that phenomenologically includes quarks confinement expected in QCD. The linear potential is the dominating part at increasing distance values, so the energy increase with distance.
The simplest case is the one in Fig. 2.14: the back-to-back production of a $q\bar{q}$ pair. The $q\bar{q}$ system evolves in space increasing the string length at some point the distance is too large and is convenient to break the string into two shorter strings.

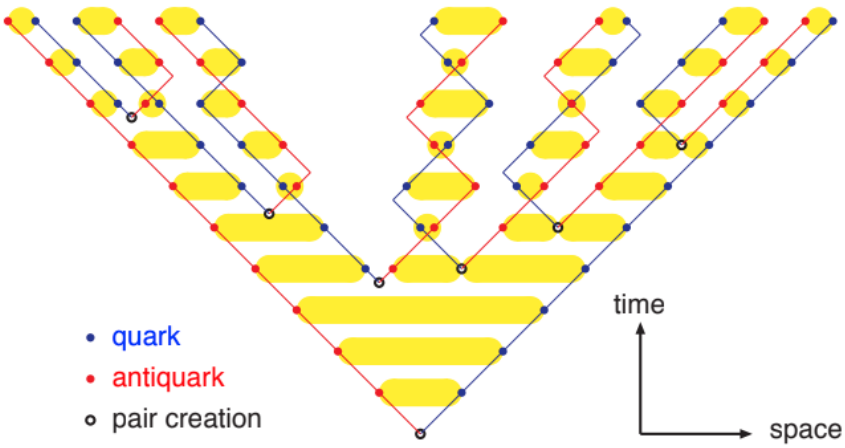


Figure 2.14: The Lund Model is schematically shown here. The string evolution in time is shown vertically, while the spatial position is displayed horizontally. As the partons move apart at some point becomes convenient to break the string, in order to reduce the total potential energy, and a partons pair is produced.

The hadronization step confines the quark into hadrons, then these hadrons can undergo to hadron rescattering and decay. These are the hadrons that are revealed by the detectors.

2.8 PYTHIA summary

To summarize PYTHIA8 is a standard tool in high energy physics studies. It is able to simulate high energy hadron-hadron collisions showing a good agreement on lots of studied distributions in the past.
The simulation starts from a hard scattering calculated perturbatively at a fixed order using matrix elements calculations. Once the hard scattering has been simulated the parton showering process takes place on all the incoming (ISR) and outgoing (FSR) partons that enter the main hard scattering.

Additionally, by the fact that hadrons are composite objects, also extra parton scatterings can occur. They are named MPI. The interplay between MPI and parton shower can be really complicated in a real collision. In PYTHIA, to achieve a good description of these complex situations the MPI, ISR and FSR are described in a common sequence of decreasing p_T starting at the hardest scale defined by the hard interaction and proceeding downward until a cutoff value, $p_{T\,0}$ is reached.
The interleaving of MPI and parton shower creates a very complex final-state of colored partons. To this final-state is then added the contribution from the primordial k_T. Moving to the real QCD case where the number of the possible charges of color is $N_C = 3$ all these colored strings, created in the previous steps, may overlap in the physical space and then reconnect each other, merging the lower-p_T MPI systems to the ones in higher-p_T MPI.
This was the last step at the partonic level simulated by PYTHIA, then the hadronization process, based on the Lund string fragmentation model takes place and transforms the partonic-state into a set of hadrons. Moreover, this hadron can undergo to hadron rescattering and decays before the detection.
All these processes simulated by PYTHIA are described mainly by phenomenological models, due to the not-known-by-first-principles soft QCD description. The use of these phenomenological models introduces lots of free parameters (some are been pointed out in these sections) that have to be tuned with data to give PYTHIA the ability to reproduce real data observed experiments. The simulation is necessary to understand the observables histograms that are collected at colliders experiments in particular for the study of the UE.

2.9 Validation and tuning of a Monte Carlo generator

Validating a Monte Carlo generator means to confront a model with all the relevant data that it claims to be able to describe. It is really important that the validation of the MC generator is global if one wants a predictive power from this generator. Otherwise, the generator is just parameterizing the data and not simulating the UE. The validation is important for developing new models and debugging physics models. Tuning the generator refers to the operation of adjusting the parameters of the generator to improve the description of the relevant data.
To perform both the validation and the tune of the generator a range of observables have to be simulated and analyzed. The various tuning computational procedure used for the optimization over the possible space of parameters are better described in Chapter 4.
A general-purpose event generator contains a lot of parameters and so the parameter space to investigate is too large, it is impossible to investigate it in a comprehensive way. The strategy is to factorize the parameters into different sets based on the group of observables has been found to be important.
So, usually, the tuning of the generators proceeds in consecutive steps:

- *Hadronization and final-state:* This step is performed employing data from e^+e^- colliders. In fact, these observables are not sensible to initial-state hadron effects. So, the typical parameters tuned in this type of event are the α_s^{FSR}

value, the cutoff of the FSR and the parameters related to the hadronization process: the string tension and the fragmentation functions.

- *initial-state shower:* Parameters related to final-state effects have been fixed. Now, the initial-state ones are investigated, typically in observables of jet shapes. The analysis of these parameters is performed before the analysis of the MPI-related ones. In this way, it is avoided the risk of absorbing effects which should be perturbatively describable into the MPI modeling. The parameters tuned in this phase are the cutoff for the ISR, the α_s^{ISR} value and others related to the initial-state effects.

- *MPI and BBR:* The parameters related to the MPI and BBR are the last tuned because they are the less constrained by the QCD calculation and have lots of free parameters that absorb all the non-perturbative effects. The typical observables for MPI and BBR tuning are underlying event data from various hadron colliders.

This thesis focus on this last step. The next chapter will focus on the study of the underlying event with the description of all the distributions used in the tune and the effects of some of these PYTHIA8 parameters on these observable distributions.

The histograms of the observables useful for testing the physics of the MC generator are encoded in the tool: *Rivet* [29].

2.9.1 Rivet and Yoda format

Rivet is a Monte Carlo event generator validation tool. It provides a large set of experimental analyses that can be used for the development, validation and tuning of event generators. Rivet structure contains different layers. A C++ shared library is the core, it is supplemented by C++ "plugin" libraries containing collider analysis routines. Then a Python programming interface and a set of Python and shell scripts are the interfaces to it.

The MC simulated events are passed into rivet that, thanks to the I/O interface based on HepMC [30] and YODA [31] libraries, perform the analysis and organize them into histograms that are saved into the YODA format. The set of the Rivet instruction to perform the analysis are usually referred to as "routine".

These YODA files are the input format for the tools used for the tune of MC event generators: PROFESSOR [8] and MCNNTUNES [7].

Chapter 3

Observables to Study the Underlying Event in CMS and Existing Tunes

The underlying events UE are all processes not associated with the primary hard scatter in a hadron-hadron collision.
All the processes described in the previous section (ISR, FSR, MPI, BBR and their interactions with color exchanges) contribute to the Underlying Event (UE) in a proton-proton collision.
As an example, in a proton-proton collision with a Z boson production, that then can decay in two leptons, l^+ and l^- observed by the experiment, the hard scattering is represented by the scattering between the two partons that generate the Z boson. While the so-called UE is represented from all the extra scatterings, the various radiations and in general all the activity not associated with this primary hard scattering.

It is important to underline the fact that most of the observables to study the UE are sensible only to the sum of these contributions and not to the single ones. So, a good description of all these processes and their interplay is really important in order to study the complex finals states originating from these scatterings that contribute to the observables.
In this chapter, these observables sensible to the UE are introduced and described with more detail but first, a description of the detector used to take the measurement is necessary.

3.1 Compact Muon Solenoid Detector

This section will briefing describe the detector of the Compact Muon Solenoid experiment (CMS). CMS is an experiment operating at the Large Hadron Collider (LHC) at CERN. LHC is a particle accelerator designed to collide protons beams. It has operated to different center of mass energies (7, 8, 13 TeV) and now with the next run called RUN3 it will operate to the record energy of 13.6 TeV. LHC is a superconducting circular accelerator with a total length of 26.7 km. It is divided into 8 sections and 4 of these are occupied by the particles physics experiments in particular one is reserved for the CMS detector.

The main feature of the CMS detector is a superconducting solenoid with a

6 m internal diameter that can provide a magnetic field up to 3.8 T. This very high magnetic field is necessary to bend the particles produced in the proton-proton interactions that take place along the beam pipe. So into the solenoid volume, there are different detectors: an all-silicon inner tracking system, a lead tungstate ($PbWO_4$) crystal electromagnetic calorimeter (ECAL), and hadron calorimeter (HCAL) composed by alternating layers of brass and scintillator material; both calorimeters are composed of a barrel, an endcap and a forward section. The outer region is covered by the muon detectors, these are composed of several layers of aluminium drift tubes in the barrel region, cathode strip chambers and resistive plate chambers in the endcap regions. An illustration is displayed in Fig. 3.1.

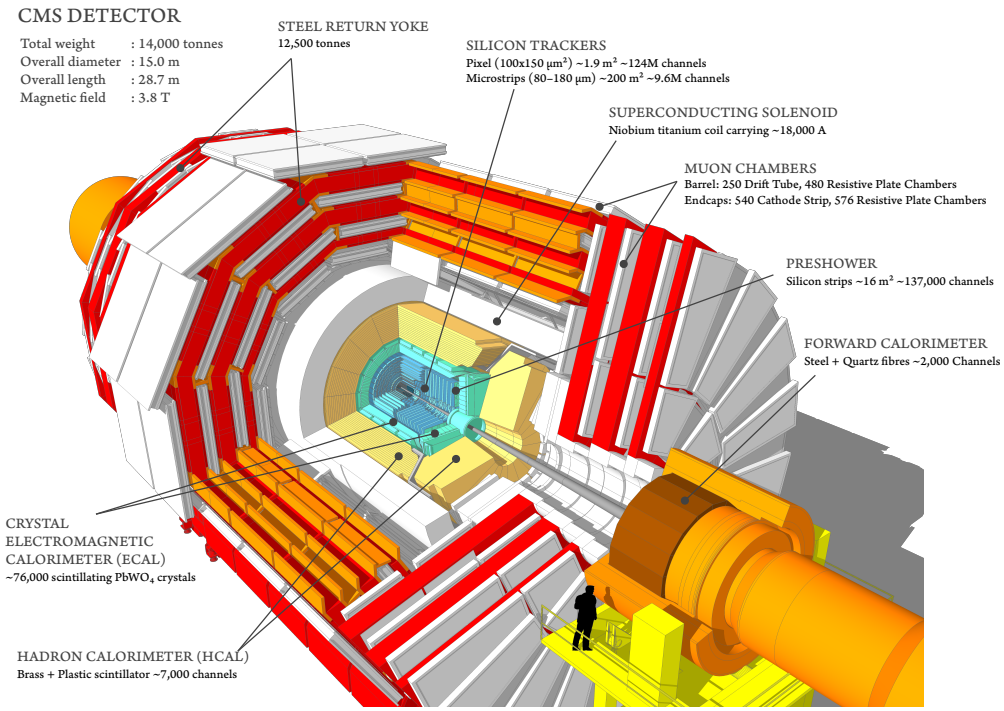


Figure 3.1: An illustration of the CMS detector [32].

The silicon tracker is responsible for the measurement of charged particle tracks within the pseudorapidity range $|\eta| < 2.5$. It is the central detector for the analysis examined in this thesis, in fact, all the the used distributions are collected by the analysis of charged particle tracks. It is obvious that a good resolution for the track reconstruction is required. The typical track resolutions, for non-isolated particles of $1 < pT < 10GeV$ and $|\eta| < 1.4$, are 1.5% in p_T. So, this inner tracking system is the central detector for the charged particles tracks reconstruction and, thanks to the high magnetic field, for the momentum measurements in CMS. The energy instead is measured by the two types of calorimeters: ECAL and HCAL.
The ECAL barrel cover the pseudorapidity range $|\eta| < 1.479$ it consists of about 61200 $PbWO_4$ crystal while the endcaps cover the pseudorapidity range $1.479 < |\eta| < 3.0$.
Instead, the HCAL barrel cover the pseudorapidity range $|\eta| < 1.3$ while the endcaps $1.3 < |\eta| < 3.0$ these colorimeters are really important in the hadronic jet detection and reconstruction.

Another main feature of the CMS detector is obviously the muon system it has three main functions: muon detection, momentum measurement and triggering, totally the muon system covers the pseudorapidity up to $|\eta| < 2.4$
The LHC provide proton-proton collision at very high interaction rates, the beam crossing interval is 25 ns. So, it is impossible to store all the data, the rate is reduced in two steps: Level-1 (L1) trigger and High-Level Trigger (HLT). These two steps perform an events selection keeping only the interest interactions. The L1 receives data from the detectors and decide if the event contains important information, for example high transverse energy particle/jets, or high missing transverse energy. The HLT , instead, is a software running on a large computer farm that receives and processes complete events with a resolution comparable to the final detector one.
In the following, the selection criteria and the space topology for the analysis used for the tunes are introduced in order to better understand what kind of observables are studied in this thesis.

3.1.1 Selection Criteria

The selection criteria for CMS data from analysis at 13 TeV [33] are the selection of charged particle tracks within $|\eta| < 2$, and $|\eta| < 0.8$ for the analyses at 7 [34] and CDF one at 1.96 TeV [35]. The trigger sensibility imposes $p_T \geq 0.5$ GeV in order to ensure a good track reconstruction efficiency. Fake tracks after mis-reconstruction are removed imposing to pass the *highPurity* [36]. Secondary decays are removed by requiring the impact parameter significance $d_0/\sigma_{d0} < 3$ and the significance in the z-direction $d_z/\sigma_{dz} < 3$. Then to keep only tracks with good resolution the tracks with relative uncertainties $\sigma_{p_T}/p_T > 0.05$ are discarded.
To reconstruct the charged particle jets the clustering algorithm used is the Seedless Infrared-Safe Cone (SISCone) [37] jet algorithm[1]. Charged particle within the pseudorapidity range $|\eta| < 2.5$ are used for jet reconstruction then the reconstructed jet have to pass the above selection criteria.
Minimum bias events are triggered by requiring activity in both Beam Scintillator Counters (BSC) that are located at a distance of ± 10.86 m from the detector center and are sensitive within $3.23 < |\eta| < 4.65$ in coincidence with signals from both beams in the Beam Pick-up Timing for eXperiments (BPTX) devices, they are located along the beam pipe at 175 m from the interaction point. These coincidences indicate that both the beams are crossing at the same time and they interact.

The selection criteria used for the pseudorapidity distributions from the analyses [39] and [40] are:

1. Online: coincident signals in both beams in the Beam Pick-up Timing for eXperiments (BPTX);

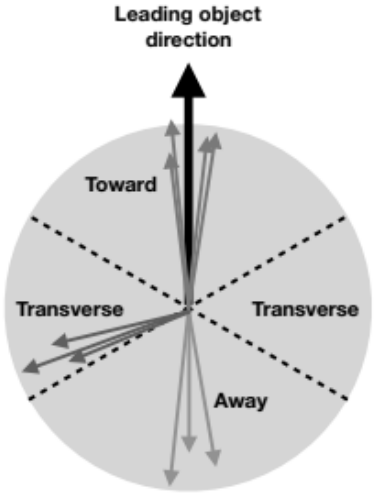
2. Offline: at least one vertex reconstruction.

The tracker measure all the charged particles within $|\eta| < 2.4$ and all the interaction points are used for the tracks reconstruction in the offline analysis.
Further details can be found in the respective paper for each analysis.

[1]The Anti$-k_T$ algorithm [38] is currently the more preferred jet clustering algorithm in CMS but to ensure continuity with respect to the previous analyses on the UE the one above is chosen.

3.2 Minimum Bias Measurements and Underlying Event topology

A minimum bias measurement is a collection of inelastic events with a loose event selection. This means that the events are collected requiring the minimal interaction with the detector (the smallest possible bias). Most of the interactions in minimum bias measurements are soft ($p_T \lesssim 2$ GeV) but the study of the UE requires at least one hard scattering ($p_T \gtrsim 2$ GeV) presence; in fact, the UE is given by the underlying activity to a primary hard scatter.
To study the UE the topological structure of a hadron-hadron collision is used. The analysis is performed on an *event-by-event* basis. In the analysis, the direction of the leading object[2] is used to define four regions in the $\eta - \phi$ space. Where η is the pseudorapidity defined as $\eta = -\log\tan\left(\frac{\theta}{2}\right)$, while ϕ is the azimuthal angle in the $x - y$ plane, which is defined from the direction of the leading object as $\Delta\phi = \phi - \phi_{\text{leading}}$.



Region	Angular region
Toward	$\lvert\Delta\phi\rvert < 60°$
Away	$60° < \Delta\phi < 120°$
Transverse	$\lvert\Delta\phi\rvert > 120°$

Figure 3.2: This figure shows the four regions for the description of the UE on an event-by-event analysis. The angular values, defining the four regions, are shown in the table. The regions are defined starting from the leading object direction. The toward and away regions contain most of the contribution from the hard scattering (e.g. in a $t\bar{t}$ production event the two quark t are located in these regions); while the transverse region are the ones that divide the toward and the away regions, these are the most important for the study of the underlying event.

The regions classification is shown in Fig. 3.2, they are divided in:

- **Toward region**: the region that contains the leading object. This region contains most of the particles produced by the hard interaction.
- **Away region**: this region contains the objects that recoil against the leading object, also this region contains most of the particles produced by the hard interaction.

[2]The leading object definition depends on the type of observable under analysis. Usually, in charged particle analyses the leading object is defined as the charged particle/charged particle jet with the largest p_T.

- **Transverse regions**: the two transverse regions are the most sensitive to UE.

The toward and away regions are the ones with the major contribution from the primary hard scattering; as an example, in a dijet event the leading jet is expected to be in the toward region while the second jet in the away region.
The two transverse regions are called:

  - **TransMAX**: This is the transverse region that contains the *maximum* number of charged particles (or scalar p_T sum of charged particles). This region includes both MPI and hard-process contamination. As an example, in events with 3-jets production this region can contain the extra jet.

  - **TransMIN**: is the transverse region that contains the *minimum* number of charged particles, or scalar-p_T sum of charged particles. This region is the most sensitive to MPI effects.

The leading object definition depends on the type of event under observation and differs from one analysis to another. The charged-particle with the largest p_T [33], the dilepton system in Drell-Yan observation [41, 42] or $t\bar{t}$ events [43] can all be used as the leading object in the analyses for the UE event.
So, the most interesting regions for the UE studies are the transMAX and transMIN regions. The main observables, that are sensitive to UE, studied in these transverse regions are the charged-particle density and the charged-particle scalar-p_T sum density in the $\eta - \phi$ space.

3.2.1 Hard scale dependence

In a hadron-hadron collision with two jets productions, it is observed that not only the charged particles activity in the toward and away regions increases with the hardness of the collision (p_T^{max}) but also the activity into transverse regions increases with it. This is shown in Fig. 3.3 where are reported the charged particle number and the charged particle p_T sum distributions as a function of the azimuthal separation with respect to the leading object ($\Delta\phi$). The different color lines refer to different energy scales for the collision. It is observed that the away region ($60 \leq |\Delta\phi| \leq 120$) at increasing energy for the collision become broader this is related to the increasing quantity of FSR radiated by the leading object.
On the other hand, the increment in the transverse regions activity cannot be explained only by the increase of the FSR from the leading object (the broadening of the away region activity peak can cause some events from the shower to end up in the transverse region). To explain this rise one needs to attribute activity in the transverse regions to MPI: the number of extra scatters increases with energy. This rise is related to the density functions for the partons inside the hadrons that increase when probed at higher energies and so the partons become denser packed in hadrons. In this way, the probability of extra scatterings is larger to higher energies.
The evolution of these two quantities in the transMAX region as a function of the transverse momentum of the leading object (measurement of the energy scale for the collision) is shown in more detail in Fig. 3.4. The two distributions related to the charged particle density and charged p_T sum in the transMAX region as a function of the leading object p_T show a rapid rise at low leading-object p_T, followed by a very slow rise.

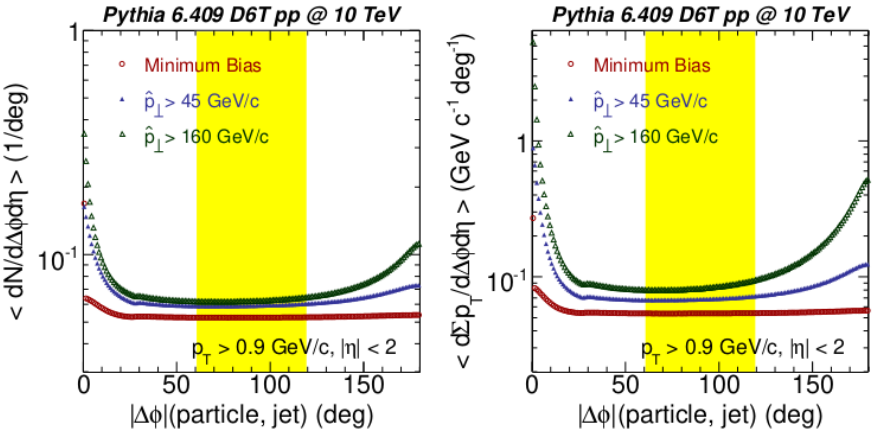


Figure 3.3: A comparison between three different scales for the interaction, in PYTHIA. The multiplicity of charged particles (left) and the scalar-p_T sum of charged particles (right) are simulated. The activity in the transverse regions increase due to two effects: the FSR is related to the broadening of the away region peak, so some events from the shower end up in the transverse region (yellow band) but this alone can't explain the increment so is required the introduction of the MPI in the description. Figure from chapter 5 of [44].

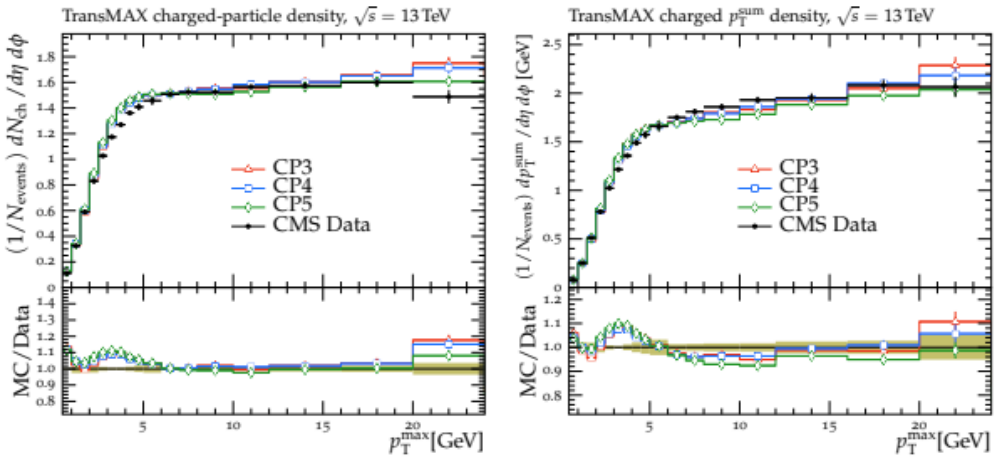


Figure 3.4: The evolution of the charged-particles density (left) and of the scalar-p_T sum density of charged particles (right) as a function of the energy scale for the scattering (leading object p_T) are shown here, the two distribution increase quite rapidly in the first bins than they saturate ($\approx 6-7$ GeV) the scalar-p_T sum density increase a little bit more, for $p_T >$ but very slowly. The black dots are the experimental point and are compared to prediction with CMS PYTHIA tunes: CP3, CP4 and CP5; they are described in more detail in the next chapter. Figure from [6].

3.2.2 Sensitivity to MPI parameters

Now the sensitivity of these observables to some PYTHIA8 parameters is investigated. Fig. 3.5 shows the effect of the MPI on both the distributions shown before separately for TransMAX and TransMIN regions. The figure shows the case with MPI (red line) and the case when the MPI are switched off (blue line). As was ex-

pected the MPI are needed to explain the activity in the transverse regions. When the MPI are switched on the number of particles produced is higher than the case when they are off. From this figure is clear that MPI are the main contribution to the activity of these TransMAX and TransMIN regions. Obviously, the discrepancy increases as the hard scale of the interaction increases, in fact in low p_T regions the p_T evolution starts at a lower value so the phase-space, for the extra collisions to occur, is smaller.

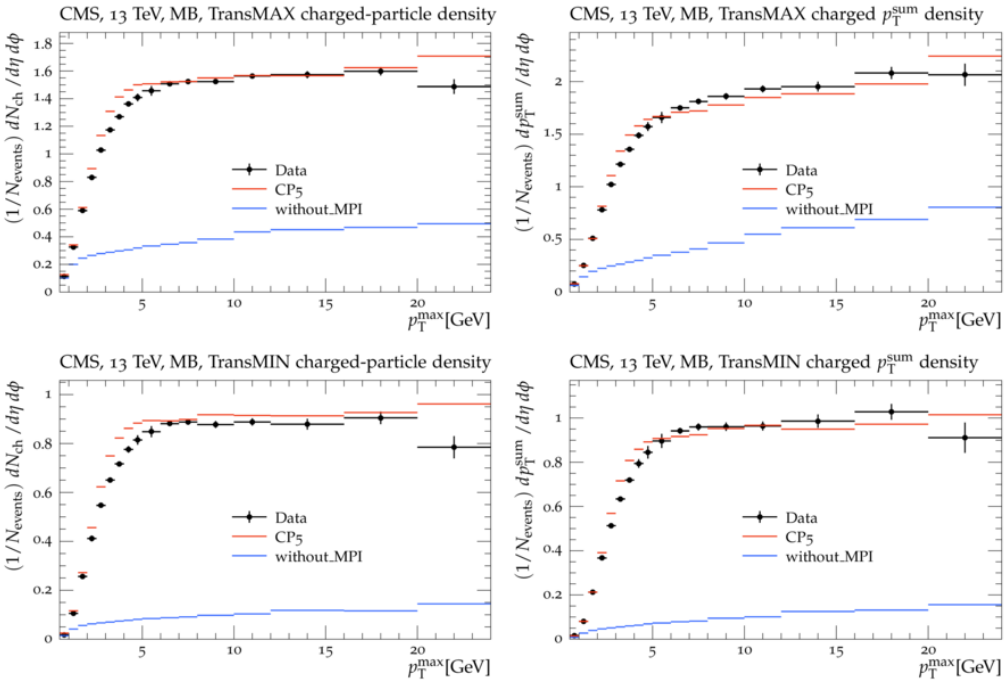


Figure 3.5: This image shows the effect of the MPI on the charged particles density and charged particles p_T sum in the transMAX and TransMIN regions distributions. The contribution of the parton shower alone cannot explain the contributions of the underlying event in these two regions (blue line) the introduction of the contribution from the MPI is necessary (red line). The two simulations are compared to the data from [33].

Another important observation that was also pointed out in the Chapter 2.5.1 is that the amount of activity in the two transverse regions is also dependent on the center-of-mass energy. This evolution is shown in Fig. 3.6 for two different energy. The amount of MPI increase with $\sqrt{s}$ that was expected from the evolution of the PDF with the energy of the collision: the hadrons become denser-packed when probed to higher energies. In Fig. 3.6 the charged particle density and the scalar sum of the transverse momentum in the transverse regions data are compared for two different center-of-mass energies (13 and 2.76 TeV). The data show that the activity in the transverse regions increase with the rise in the center-of-mass energy. The data are also compared to existing different CMS tuning.

The data from the activity in the TransMAX and TransMIN regions have been collected by a large number of experiments at different center-of-mass energies. So these observables related to the charged particles activity in the two transverse regions at various center-of-mass energies are the main ones that will be used in the

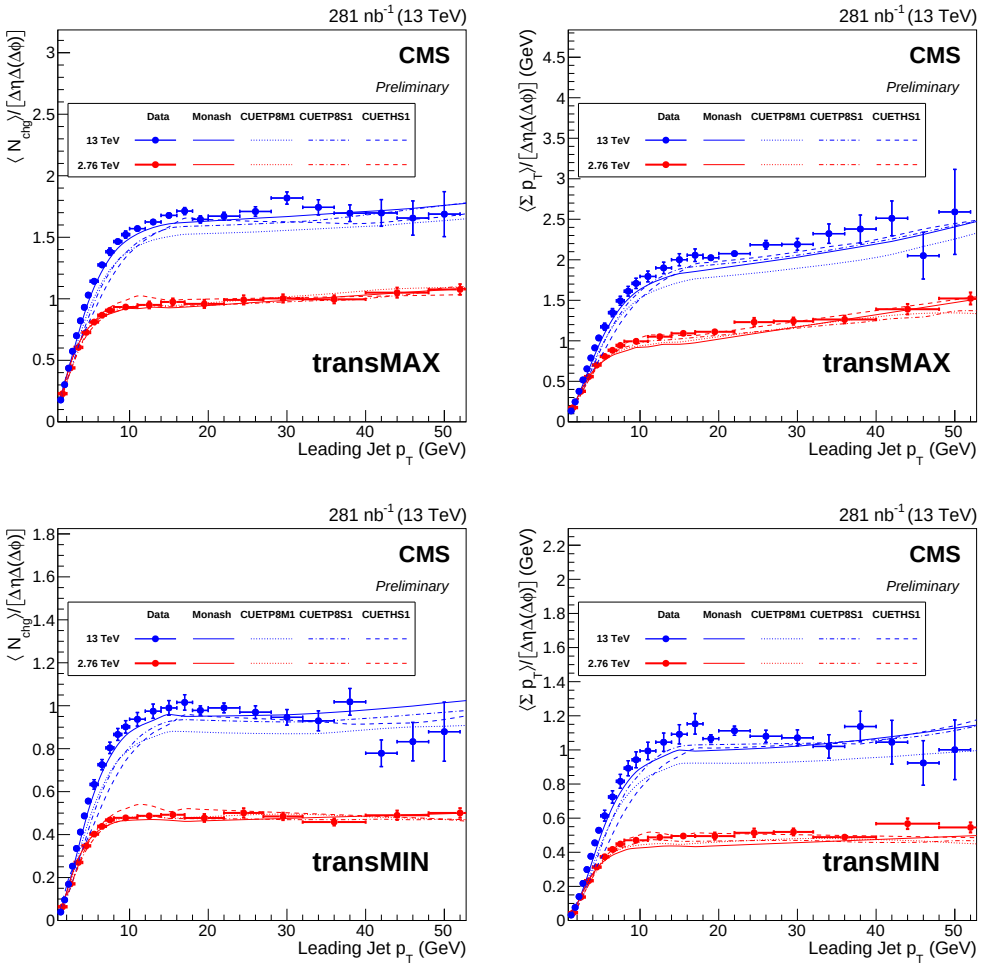


Figure 3.6: The charged particle density in the transMAX (upper left) and transMIN (lower left) regions and of the charged particle p_T-sum in the transMAX (upper right) and the transMIN (lower right) regions evolution as function of the center-of-mass energy is shown. The red ones are the data for $\sqrt{s} = 2.76$ and the blue ones for $\sqrt{s} = 13$ and these are compare to different CMS tunes. Figure from [33]

tunes presented here and used in previous tunes for the UE.

3.2.3 Observables in Z production processes

In the last part of this work, it will performed the tune of primordial k_T another important parameter in the simulation of hadron-hadron collisions. An open question is that the primordial k_T value required for the description of the observed cross-section for the Z production is very large with respect to the expected value derived in Eq. 2.45 by the typical size of a hadron. This very high value is not understood by first principles. So, it is required a tune for this parameter in the Monte Carlo generator in order to describe the experimental data in a good way.
The observables to study the primordial k_T are the differential cross-section for Z production in proton-proton collision in Drell-Yan observation as a function of the transverse momentum of the Z boson (Fig. 3.8 left) and as a function of the ϕ^*_η angle

(Fig. 3.8 right), where the angle ϕ^*_η is defined as:

$$\phi^*_\eta = \tan\left(\frac{\pi - \Delta\phi}{2}\right) \sin\theta^*_\eta \quad , \tag{3.1}$$

where $\Delta\phi$ is the azimuthal angle between the two leptons. The angle θ^*_η instead is the angle measured with respect to the beam direction in the rest frame of the dilepton system (frame with leptons emitted back-to-back), so is defined as:

$$\cos\theta^*_\eta = \tanh\left(\frac{\eta^- - \eta^+}{2}\right) \quad , \tag{3.2}$$

with η^+ and η^- the pseudorapidities of the two leptons [45], a schematic view of the angles defined in the interaction is displayed in Fig. 3.7.

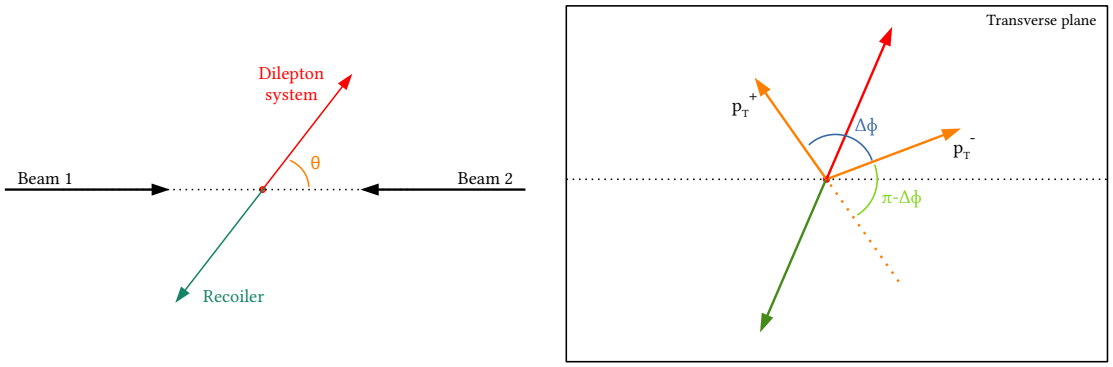


Figure 3.7: A schematic representation of a Drell-Yan observation with a description of the angles used in the definition of ϕ^*_η in Eq. 3.1.

The Z boson is a very massive boson, $m = 91.1876 \pm 0.002$ GeV [2], which can be produced only in the hard scattering of two partons of the hadron undergoing the collision, then after a very short lifetime the Z boson decay in a couple of leptons: $e^+\,e^-$ or $\mu^+\,\mu^-$. The measurement of this final state takes information on the Z boson that is not directly observed by experiments. It is a matter of fact that the production of the Z boson can be affected by all the other soft processes taking place along with this production that can change the topology of the initial state. In the last chapter it is investigated the effect of ISR and Primordial k_T on the Z boson production in the low region of the spectrum.
Since the Z boson production is a hard process to full simulate the Z production spectrum it is necessary to merge matrix element calculation and space shower to do this it is employed the use of a margin scheme. To have a good description of the high region of the spectrum ($p_T^Z > 2$ GeV) one have to include higher-order calculations (NLO and NNLO processes) and, consequentially, a more complex merging scheme as the FxFx merging scheme that has been introduced in Chapter 2.
As shown in Fig. 3.8 the description at LO (and without the need for FxFx) is good only in the first bins that are the ones of interest for the study of underlying event effects and it is less computationally expensive than running all the FxFx merging scheme. So, the strategy followed for the tuning of the primordial kT in the Chapter 6 is to tune the Primordial k_T in this low region, the one of interest, taking only the first 5 bins from each distribution.

Then the full simulation with higher-order calculations and the use of FxFx margin scheme is run only for the comparison once the parameters have been tuned. This is going to be discussed in more detail in Chapter 6 where the primordial k_T tune is performed.

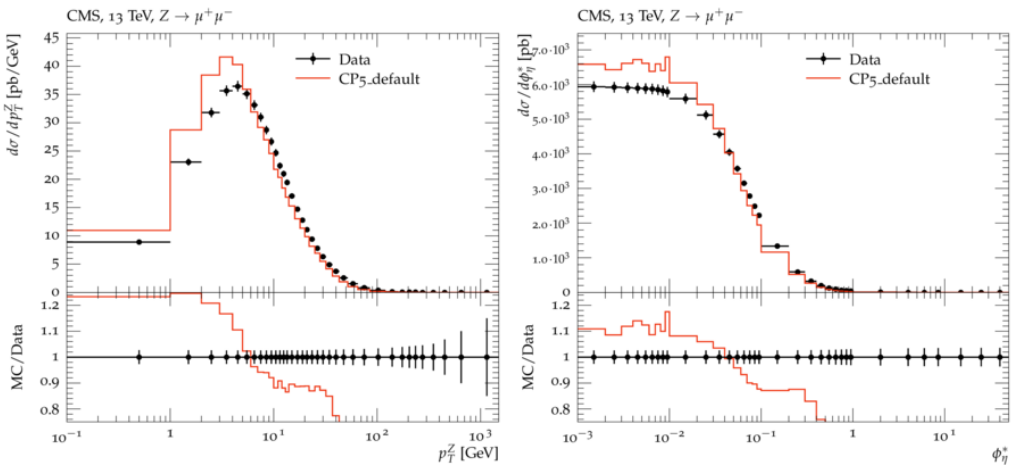


Figure 3.8: The distributions shown are from [46] CMS analysis at $\sqrt{s} = 13$ TeV, these will be used for the tune of the primordial k_T. The left one is the Z boson production cross-section as a function of the transverse momentum of the Z boson. On the right the differential cross-section for the Z boson production as a function of the ϕ_η^* angle. The red line is the PYTHIA8 description before the tune of the primordial k_T and without the FxFx merging scheme so only the first bins have to be taken into account.

The following section will introduce the existing CMS tunes for the UE in minimum bias observables. These tunes are able to reproduce very well the observables described in Section 3.2 at 1.96, 7 and 13 TeV.

3.3 Previous Tune for the Underlying Event

In [6] the CMS collaboration presents new PYTHIA8 tunes for the underlying event (UE). The tunes are called CP and a number from 1 to 5 where CP stands for "CMS PYTHIA8". The tunes are performed by investigating the effects of different values of the coupling constant α_s for the ISR, FSR, hard scattering and MPI and changing the order of the evolution for the α_s with the Q^2 value of the interaction. Another difference among these tunes is the choice of the PDF set: CP1 and CP2 use a LO PDF, CP3 an NLO one, while CP4 and CP5 tunes use an NNLO PDF set.

The tuned parameters are shown in Table 3.1 with the associated variation ranges and a recall of the definition of each one.

Parameter description	Name in PYTHIA8	Range considered
MPI threshold [GeV], `pT0Ref`, at $\sqrt{s} = \sqrt{s_0}$	`MultipartonInteractions:pT0Ref`	$1.0 - 3.0$
Exponent of $\sqrt{s}$ dependence, ϵ	`MultipartonInteractions:ecmPow`	$0.0 - 0.3$
Matter fraction contained in the core	`MultipartonInteractions:coreFraction`	$0.1 - 0.95$
Radius of the core	`MultipartonInteractions:coreRadius`	$0.1 - 0.8$
Range of color reconnection probability	`ColorReconnection:range`	$1.0 - 9.0$

Table 3.1: This table reports the five parameters tuned for the UE in CP* tunes, the variation ranges used for the sampling are shown in the last column. Table from [6]

These parameters are the ones that govern the number of MPI and the amount of color reconnections[3]. These CP* aims to be general multi-purpose tunes for UE and minimum bias observables described above.
These tunes are performed with the standard tool for the high energy physics tuning: PROFESSOR.
The tunes are performed investigating the five-dimensional space, reported in Table 3.1, with around 150 different choices of the five parameter values.

3.3.1 The distributions used for the Tune

The observables distributions used for the tune are the following ones:

- The pseudorapidity distribution of charged hadrons (p, K ,π) measured for an inclusive selection in inelastic ND proton-proton collisions [40];
- Charged particle density and charged particle scalar p_T^{sum} in TransMIN and TransMAX regions at different $\sqrt{s}$: 1.96 TeV [35], 7 TeV [34], 13 TeV [33];
- The pseudorapidity distributions for single diffractive (SD) and non single diffractive (NSD) events selection [39].

In the next chapter, the tune using MCNNTUNES will be described and to perform it the same distribution listed here are used.

3.3.2 Pyhtia configuration and the tunes

On the top section of Table 3.2 and Table 3.3 are reported the values for the PYTHIA8 parameters used in the CP tunes and, on the bottom, the five parameters resulting from the tune.
CP5 is the main tune on which this thesis focuses. The aim of the first part of this thesis is to reproduce this tune using the same settings for PYTHIA8 but using a different tuning software: MCNNTUNES.

The next chapter discusses the general approaches to the tune, gives the reason why one want to use the machine-learning-based approach and a description of the main tool used for the tuning in this thesis MCNNTUNES.

[3]Note that all the other parameters are set to the values obtained by the Monash 2013 tune [47].

PYTHIA8 parameter	CP1	CP2
PDF Set	NNPDF3.1 LO	NNPDF3.1 LO
$\alpha_S(m_Z)$	0.130	0.130
`SpaceShower:rapidityOrder`	off	off
`MultipartonInteractions:EcmRef` [GeV]	7000	7000
$\alpha_S^{\mathrm{ISR}}(m_Z)$ value/order	0.1365/LO	0.130/LO
$\alpha_S^{\mathrm{FSR}}(m_Z)$ value/order	0.1365/LO	0.130/LO
$\alpha_S^{\mathrm{MPI}}(m_Z)$ value/order	0.130/LO	0.130/LO
$\alpha_S^{\mathrm{ME}}(m_Z)$ value/order	0.130/LO	0.130/LO
`MultipartonInteractions:pT0Ref` [GeV]	2.4	2.3
`MultipartonInteractions:ecmPow`	0.15	0.14
`MultipartonInteractions:coreRadius`	0.54	0.38
`MultipartonInteractions:coreFraction`	0.68	0.33
`ColorReconnection:range`	2.63	2.32

Table 3.2: CP1 and CP2 tunes settings are reported here together with the values for the parameters tuned. CP1 and CP2 use a LO PDF set. CP1 α_s is different between matrix element calculation and MPI that use a value of 0.1365 ISR and FSR that instead uses 0.130. While CP2 use the same value for all processes, it is fixed at 0.130. In both cases α_s run with a LO evolution. Table from [6]

PYTHIA8 parameter	CP3	CP4	CP5
PDF Set	NNPDF3.1 NLO	NNPDF3.1 NNLO	NNPDF3.1 NNLO
$\alpha_S(m_Z)$	0.118	0.118	0.118
`SpaceShower:rapidityOrder`	off	off	on
`MultipartonInteractions:EcmRef` [GeV]	7000	7000	7000
$\alpha_S^{\mathrm{ISR}}(m_Z)$ value/order	0.118/NLO	0.118/NLO	0.118/NLO
$\alpha_S^{\mathrm{FSR}}(m_Z)$ value/order	0.118/NLO	0.118/NLO	0.118/NLO
$\alpha_S^{\mathrm{MPI}}(m_Z)$ value/order	0.118/NLO	0.118/NLO	0.118/NLO
$\alpha_S^{\mathrm{ME}}(m_Z)$ value/order	0.118/NLO	0.118/NLO	0.118/NLO
`MultipartonInteractions:pT0Ref` [GeV]	1.52	1.48	1.41
`MultipartonInteractions:ecmPow`	0.02	0.02	0.03
`MultipartonInteractions:coreRadius`	0.54	0.60	0.76
`MultipartonInteractions:coreFraction`	0.39	0.30	0.63
`ColorReconnection:range`	4.73	5.61	5.18

Table 3.3: Here are reported CP3, CP4 and CP5 tunes settings, and the results for the tune. The three tunes use an equal α_s value for all the processes, $\alpha_s = 0.118$ running with a NLO evolution. The difference between CP3 and the other two tune is that CP3 use an NLO PDF set while CP4 and CP5 an NNLO one. CP5 ISR emission is also ordered according to rapidity. Table from [6]

Chapter 4

Tuning procedure and MCNNTUNES

As was introduced in the first chapter the tuning procedure can be really computational expensive, in fact it requires to run the generator a very large amount of times, and usually these MC generators are really expansive in terms of computational time required for just a single job.
To tune a certain number of parameters the number of Monte Carlo runs needed for the tuning procedure increases with the number of free parameters. In fact, the dimension of the parameter space increases with the number of the sampling one wants to perform and the granularity one wants to achieve. So, in the past, different approaches have been developed and used to tune these MC generators. A brief description is reported for each approach in the following list:

1) **Manual tunes**: this approach is based on an optimization of the parameters made by eye. This is absolutely not the best way to tune some parameters, usually it requires a very large time for even semi-reasonable results since the process requires a very large number of iterations.

2) **Brute force tunes**: a better way would be to perform a very dense sampling in parameters space, run the generator with every configuration and then chose the configuration corresponding to the output that better describes the experimental data. This is very computationally expensive and a scan in a 5 parameters space with 10 divisions each requires $10^5 = 100000$ Monte Carlo runs this with a rising number of parameters becomes really impractical, also using computers batch systems as an example HTCondor.

3) **Parametrization-based tunes**: an even more better approach is to find a surrogate function to parameterize the response of the MC generator at different values of the parameters to tune, and try to study (minimize) this surrogate function instead of the real response of the generator. This is the current approach that is used in the high energy physics tune and that will be used in the following tunes presented in this thesis.

This last approach will be discussed in more detail in the next section.

4.1 Parametrization-based approach

The parametrization-based approach is the most used method. The current state-of-art in the tuning procedure is to use a polynomial function to fit the response of the generator. Once the parametrization is performed the tuned parameters are given by the minimization of this parameterized response function. This approach based on the polynomial parametrization has been implemented in the software PROFESSOR [8].
So the first step in the procedure is to fit the response of the generator using a surrogate function simpler to study than the real one (e.g. Professor instead of the arbitrary complex real function uses a polynomial fitted to well describe the real function).

$$h(p) \xrightarrow{\text{parametrization}} \overline{h}(p) \quad , \tag{4.1}$$

where $h(p)$ is the vector of the observables distribution bins from MC simulations that are a function of the parameters. In this phase, these functions that give the values of the bins at the variation of the parameter values have been substituted by the surrogate ones $\overline{h}(p)$.
After that, a *loss function* $\mathcal{L}(\overline{h}(p), h_{\text{data}})$ is defined between the surrogate function and the experimental data. A common choice for it is the χ^2 function defined as:

$$\mathcal{L}(\overline{h}(p), h_{\text{data}}) \equiv \chi^2 = \frac{(\overline{h}(p) - h_{\text{data}})^2}{\sigma^2} \quad . \tag{4.2}$$

where σ^2 is the vector of the experimental squared error on each bin. In the end, to find the best parameters estimation, this loss function needs to be minimized. The set of parameters p_{best} is the best evaluation that the generator can provide for the real values and this set of best parameters is going to be referred to as *tune*.

$$p_{\text{best}} = \arg\min_{p} \ \mathcal{L}(\overline{h}(p), h_{\text{data}}) \quad . \tag{4.3}$$

In this thesis instead of the common software PROFESSOR based on the polynomial parameterization, it is used the machine learning approach implemented in MCNNTUNES software [48] using Feed-Forward Neural Networks. MCNNTUNES is a software developed by S. Carrazza, S. Alioli and M. Lazzarin presented in [7] based on machine learning library TensorFlow [49]. MCNNTUNES is written in Python and uses Feed Forward Neural Networks that are trained in order to learn the generator behaviour with respect to the parameter variations. This removes the polynomial constraint in the fit of the generator response function; in fact, one of the main features of the Neural Networks (NNs) is that they are universal function approximators.
A brief introduction follows on machine learning and in particular on NNs in order to understand better how MCNNTUNES works.

4.2 Machine Learning and Neural Networks

Machine learning (ML) is a particular type of Artificial Intelligence, it consists of the implementation of systems that learn automatically by the data that are fed

into them and not by the explicit programming of the algorithm. ML requires the training of the algorithm in order to have a predictive output related to the problem under analysis. The training is the most important step in the ML approach, in fact, this is the step that gives to the ML the ability to return a predictive output; without it, the outcome of the machine learning algorithm is not able to give us a meaningful result.
A particular type of ML is Deep Learning which uses neural networks with more than one layer organized in a hierarchical structure to solve the problem. This is the type of ML interesting for this work. In the next section the simple possible "neural network" that can be created is explained, this is useful to understand the basis in the logic of each unit that is going to compose the network.

4.2.1 Neural Networks - Perceptron

The concept of the Neural Network was developed in 1958 by Frank Rosenblatt. He introduces the simpler example of NN: the perceptron [50]. A representation of a percepton is shown in Fig. 4.1 the input values are weighted and summed, an additional offset b can be introduced, and then the weighted-sum is passed to an activation function (step function). So, the output that the perceptron returns is:

$$h(x) = \text{step}\left(\sum_j w^j x_j + b\right) \tag{4.4}$$

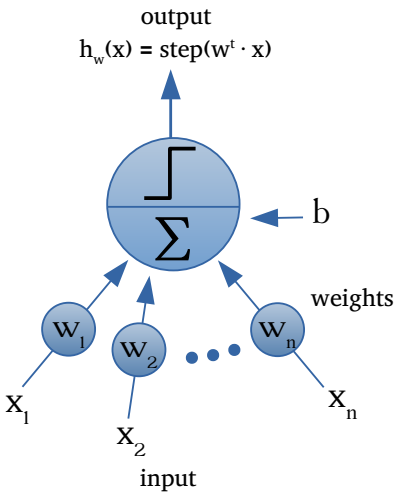


Figure 4.1: A schematic representation of a perceptron.

The revolutionary feature of the perceptron was the ability of learning by an adjustment of the weights. Anyway, a single perceptron is not enough since these kinds of logical units have lots of limitations. An example of a limitation for the perceptron is shown in Fig. 4.2 where the impossibility of implementing a XOR operation using a perceptron is shown with a graphical explanation. The perceptron is a linear classification algorithm and in the image is represented as a blue line that set a boundary for the acceptation of the hypothesis.

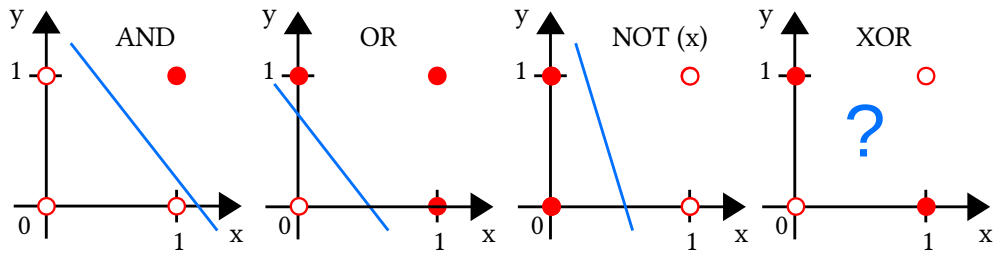


Figure 4.2: The figure shows one of the limitations of the perceptron. The XOR operation is not possible with a linear cut. In the figure the two axes are related to the two boolean variables x and y; the possible values for these variables are only 0 and 1, the four combinations are indicated by the red circles: $(0,0)$, $(1,0)$, $(0,1)$ and $(1,1)$. The first image shows the AND operation the full red circle indicates the accepted point $(1,1)$, the discrimination can be performed correctly with just one straight line (perceptron discrimination). Also, the OR ($(0,1)$, $(1,0)$, $(1,1)$) and the NOT X ($(0,0)$, $(0,1)$) ones can be implemented correctly using perceptron. But this cannot be done for the XOR operation ($(0,1)$, $(1,0)$).

To describe more complex problems, the concept of NN with more than one unit has been introduced: this is necessary if one wants to eliminate these limitations and approximate every type of function.

4.2.2 Feed-Forward Neural Networks

In a Neural Network different units called "neurons" are linked together. Different types of NNs exist and they are classified according to the various ways the neurons are linked to each other. *fully-connected feed-forward NNs* are the ones of interest, since are the ones used in MCNNTUNES.
In fully-connected NNs each neuron from a layer is connected to every neuron in the next layer, while the Feed Forward attribute refers to the fact that the NN have no internal recursions (loops) between neurons but all the neurons from a layer are connected forward to the ones of the next layer.
Fig. 4.3 shows a schematic view of a fully-connected feed-forward multi-layer NN. The basic idea is that the neurons can get some value in input and return a value as output. The output of a neuron is then sent to all the neurons in the next layer and weighted differently for each one. To be more specific: each unit j in the hidden layer (i) takes a vector of values x^k, that come from all the k neurons of the previous layer, and an offset ϑ_j as input, computes the weighted sum $\sum_k w_{kj}x^k$ and apply the activation function ϕ to the result. Common choices for this function are *tanh* or *sigmoid* functions. So, the total output of the (i)-th layer in the network is the function:

$$f^{(i)}(x) = \sum_{j=1}^{N^{(i)}} \varphi\left(\sum_k w_{kj}^{(i)} x^k + \vartheta_j^{(i)}\right) \quad , \tag{4.5}$$

where $N^{(i)}$ is the number of hidden units in the (i)-th layer.
One of the prominent features of the NNs is that they are universal function approximators [51, 52] the only request for this is that a sufficient number of hidden

layers is available. This is the main reason to use a NN-based approach instead of the polynomial one implemented in PROFESSOR.

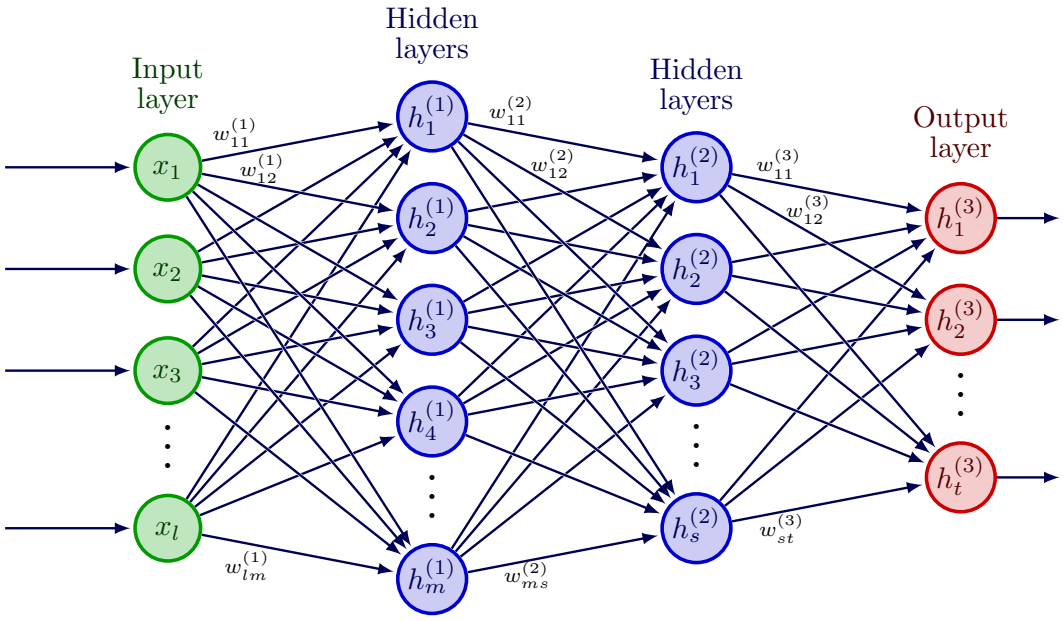


Figure 4.3: A fully-connected feed-forward neural network with more than one hidden layer, the green one is the input layer and takes the value in input and usually scale them in the range $[0, 1]$ and then the data sequentially go trough the neurons in the hidden layers (blue) where all the steps described above are performed for each layer. At the end the results of the computation are collected by the output layer (red).

As mentioned before the main feature of all the types of NNs is the ability to learn from data without being directly programmed. But to this and so get some predictive results from the NN a learning algorithm has to be defined.
A common training algorithm for the NN is the *back-propagation*, where a set of Monte Carlo simulations is used to train the NN. The back-propagation procedure is based on the idea of changing the weights, w_{jk}, and the offsets, ϑ_j, in order to minimize a loss function usually defined as the mean squared error (E):

$$E = \frac{1}{2}\sum_i (h_i(x^j, w_{jk}) - d_i)^2 \quad , \tag{4.6}$$

where the h_i are the value in output from the NN and the d_i the real value known from the Monte Carlo truth.
In the back-propagation algorithm the weight and the coefficients are updated using the *steepest-descent minimization*:

$$w_{jk}^{(i+1)} = w_{jk}^{(i)} - \lambda \left(\frac{\partial E}{\partial w_{jk}}\right)^{(i)} \quad ; \qquad \vartheta_j^{(i+1)} = \vartheta_j^{(i)} - \lambda \left(\frac{\partial E}{\partial \vartheta_j}\right)^{(i)} \tag{4.7}$$

where λ is the *learning rate*, it is a user-tunable free parameter and controls how much the weights change each time they are updated. It is one of the main parameters in the neural networks a too small value for it can lead to a failure in the

training procedure while a too large one can lead to unstable results.
The *mini-batches* are subsets of the training set that contains the Monte Carlo simulations that are used to calculate the gradient. The size of the mini-batches used to train the NN is also a free parameter: smaller batches are faster to compute but the gradient direction is not the real one just an approximation; while a bigger batch size gives a good approximation of the direction of the steepest-descent but can be computationally expensive.
The number of evaluations of the entire training set is called *epoch* and it is tunable by the user.
Note that the batch size and the number of epochs are strictly related. A training set of 50 MC runs with a batch size of 10 and a number of epochs of 1000 requires 5000 iterations, while if a batch size of 25 is used it requires only 2000 iterations to run over the whole training set.
A schematic representation of the training process is displayed in Fig. 4.4 where the training set is subdivided into the mini-batches that one by one are fed to the NN. Then, the Monte Carlo truth and the output of the network are used to compute the loss function E, using this information the back-propagation, based on the gradient descend algorithm, updates the weights in the descent direction of the gradient calculated using the mini-batch. All the procedure is repeated with every mini-batch and for a user tunable number of epochs.

Note that the prediction capability of the Neural Network is dependent on the selection of all these user-tunable parameters that control the Neural Network architecture. All these parameters are usually referred to as *hyperparameters*.

In the next section, MCNNTUNES is introduced and all its working modes are explained in detail.

4.3 MCNNTUNES

MCNNTUNES [7] is a Shower Monte Carlo generators tuning tool that implements a tune procedure based on the use of Feed Forward Neural Networks. The advantage of using FFNNs has been described above and it is that they are universal function approximators at the simple cost of having a sufficient number of hidden units in the hidden layers. This feature allows us to remove the polynomial bias present in PROFESSOR tool.
MCNNTUNES offers two different main operation modes: *PerBin Model* and *Inverse Model*. The first one is based on an approach similar to the one in PROFESSOR but where the response of the generator is parameterized using these FFNNs, one for each bin; the latter is a totally new approach where the NN (only one in this case) is trained to learn the inverse function of the generator response and then used to tries to infer the parameters value starting from the experimental values of the bins in the distributions used.

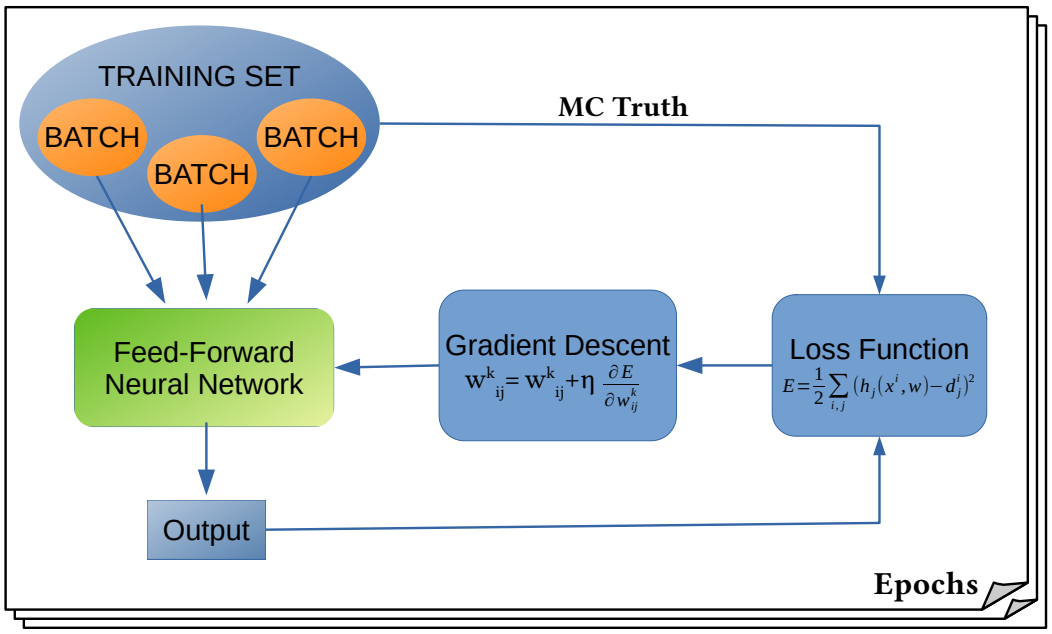


Figure 4.4: The training set used to train the NN is divided into mini-batch (the mini-batch size is a user tunable parameter) then one by one the batches are fed to the NN the output of the network together with the Monte Carlo truth are used to calculate the loss function. Then the weights and the offset are updated as described in Eq. 4.7 with the back-propagation algorithm. This is done with each mini-batch in the training set and for a number of epochs defined by the user.

4.3.1 Sampling Phase and Training Set Generation

The two approaches have the same starting point that is a sampling of the parameter space (e.g. for the UE analysis the parameters space shown in Table 3.1 has been used), then the generator is run with every sampled configuration. All these MC runs are going to build the dataset which is called *training set*.
Fig. 4.5 shows a schematic explanation of how the sampling phase and the training set generation were performed using the CMS software environment referred as CMSSW. CMSSW is present on the public machine cluster of the CERN called LXPLUS (LinuX Public Login User Service). In more detail the sampling starts by means of the MCNNTUNES script called MCNNTEMPLATE with a sampling of the defined parameter space. The sampling generates N different configurations for the parameters to tune, these are then encapsulated in a runcard for PYTHIA that contains all the necessary information to make PYTHIA work properly (beams energies, type of event to simulate etc.). Once these runcard are saved the PYTHIA generator is run with every configuration generated and this is performed not locally but on a computers batch (e.g. CondorHT) in order to gets all the results in few days. The runcards passed to CondorHT contain also the information on the analysis to perform and on how to fill the various histograms for the observables. Once the generators runs end the outputs are saved in the YODA format [31], that is a standard type of data file for these analyses also used in Rivet. The set of these YODA files, containing the information on the MC runs, composes the training set that then can

be used for the tuning procedure.

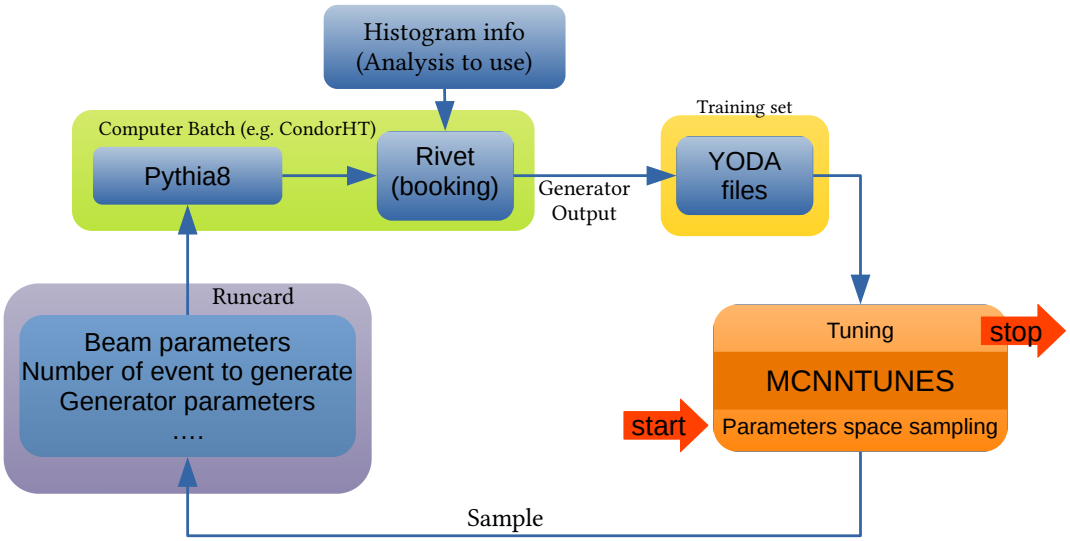


Figure 4.5: A schematic description of the main step from the sampling phase to the generation of the training set used for the tune procedure. These shows how the work was performed in CMSSW environment. The starting point and the final tune procedure are both controlled by MCNNTUNES. In the middle of these two phases the generation of the training set is required, this is performed running the generator many times.

MCNNTUNES offers also the possibility of changing the value of the hyperparameters. It is possible to choose the NN architecture: the number of hidden layers, the number of neurons for each layer and the activation function used. It is also possible to set the number of epochs, the batch size and the learning rate in order to have the best train for the architecture selected and other choices are possible. This is really important in order to get the best possible results for the tune. Below, the procedure used for the Inverse Model to search for the best hyperparameters configuration will be discussed.

4.3.2 PerBin Model

PerBin Model is a parametrisation-based method. The main idea, as shown in Fig. 4.6 is to build a model (i.e. a neural network) for each bin in order to parameterize the generator output. Each NN takes the parameter values as input and returns the bin value as output.
All these NNs are then trained feeding the MC runs from the training set and using a gradient-based algorithm, as usual for feed forward neural networks, with mean squared errors as loss function.
Once, the NN is trained, the last step is the tune in which one actually gets the best parameters estimation. This step defines a surrogate loss function for the tuning problem. In fact, the parameterization step returns a model $h^{(i)}(\mathbf{p})$ for each bin, i, where $\mathbf{p}$ is the vector of the parameters.

Then, this surrogate loss function that is defined as:

$$\chi^2 = \sum_{i=1}^{N} \frac{\left(h^{(i)}(\mathbf{p}) - h^{(i)}_{exp}\right)^2}{\sigma^2_{(i)}} \tag{4.8}$$

need to be minimized in order to evaluate the best estimation for the parameters. In MCNNTUNES this minimization is performed using the CMA-ES algorithm [53]. So the best estimation for the parameters is the configuration of parameters that minimizes this χ^2.

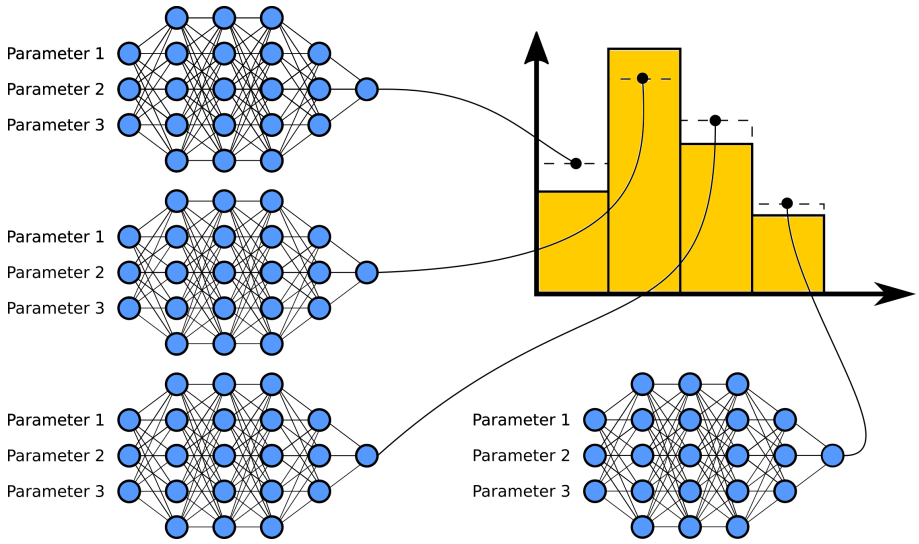


Figure 4.6: Figure from [7]

Error evaluation

MCNNTUNES PerBin model as introduced in [7] did not have a proper error evaluation on the minimized parameters. In fact it was using as error the final width of the distribution of sampled point in the CMA-ES algorithm. But, this was not working the errors were underestimated on all the minimized parameters. Thanks to S. Carrazza and with the help of M. Lazzarin it has been possible to handle the code and add a proper error estimation.

Now, the error evaluation for the PerBin model is given by the definition of a confidence interval using the χ^2 function. In fact, as shown in section 9.6 and 9.7 of [54], for an estimators vector $\hat{h}(\mathbf{p}) = (\hat{h}^{(1)}(\mathbf{p}), \hat{h}^{(2)}(\mathbf{p}), \ldots, \hat{h}^{(n)}(\mathbf{p}))$ for the parameters $\mathbf{p}$ the probability distribution function and the likelihood (the χ^2 in this case) in limit of a large set of estimators and so of degrees of freedom (ν) is approximately Gaussian distributed[1] with mean ν and variance 2ν. The probability distribution function for the estimators is then:

$$f(\hat{h}(\mathbf{p})|h(\mathbf{p})) = \frac{1}{(2\pi)^{n/2}|V|^{1/2}} \exp\left[-\frac{1}{2}\left(\hat{h}(\mathbf{p}) - h(\mathbf{p})\right)^T V^{-1} \left(\hat{h}(\mathbf{p}) - h(\mathbf{p})\right)\right], \tag{4.9}$$

[1] In other cases this is a good approximation.

percentile	Q_γ				
	$\nu = 1$	$\nu = 2$	$\nu = 3$	$\nu = 4$	$\nu = 5$
0.683	1.00	2.30	3.53	4.72	5.89
0.90	2.71	4.61	6.25	7.82	9.24
0.95	3.84	5.99	7.82	9.49	11.1
0.99	6.63	9.21	11.3	13.3	15.1

Table 4.1: The table report the values of the quantile Q_γ for different percentile and for different number of degree of freedom ν. The row corresponding to 0.683 (1σ) is the one of interest here.

where T is the transposed vector and V^{-1} is the inverse covariance matrix. So, the confidence interval can be defined using the χ^2 statistic as:

$$\frac{\chi^2(\text{c.i.})}{N_{dof}} = \frac{\chi^2_{min}}{N_{dof}} + \frac{Q_\gamma}{N_{dof}} \tag{4.10}$$

The variation is dependent on the number of parameters and on the chosen confidence level ($1\sigma = 0.683$ in this case) and a list of the values is reported in Table 4.1. Then, the error is defined as the value of the parameters that give a deviation from the minimum value of the χ^2/N_{dof} equal to the Q_γ/N_{dof} value for a confidence level of 0.683, as defined in Eq. 4.10.
An example for the evaluation in MCNNTUNES is shown in Fig. 4.7 where the green line is the deviation from the minimum value of the χ^2 defined in Eq. 4.10 and the errors are given by the points where the χ^2/DoF reaches these values, in the figure given by the intersection of the blue and the green line.

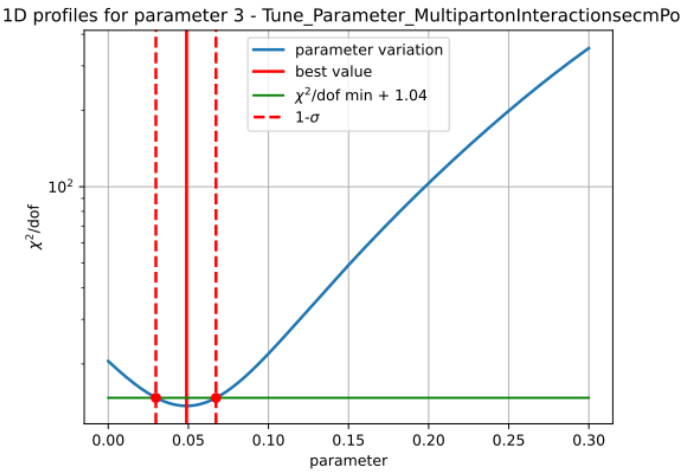


Figure 4.7: This figure shows the error evaluation in MCNNTUNES for the `MultipartonInteractions:ecmPow Pythia8` parameter. The blue line is the value of the χ^2/N_{dof} evaluated for different parameter values. The best estimation is indicated by the vertical red line, while the green line is the quantity in Eq. 4.10. The error is evaluated from the intersection of blue and green lines.

4.3.3 Negative aspects

One of the negative aspects is that this model is more computationally expensive than the Inverse model discussed below. This is due to the large number of NNs built and trained from this model. The high cost in terms of time required to get the model work do not give the possibility for a scan in the hyperparameters space in order to obtain the best configuration for the NN architecture.
This can impose some limitations on the model performance that cannot reach its maximum potential.

4.3.4 Inverse Model

The Inverse Model is the most innovative tuning procedure introduced by MCNNTUNES. This model contrarily to the PerBin Model takes the bins of the histograms as input and returns parameter values as output. For the Inverse Model the NN used is only one as shown schematically in Fig. 4.8. What the Inverse Model tries to do is to learn the inverted model of the generator. So starting from the observed values the model tries to reproduce the parameters values necessary to get the histograms used as input.
The model is built and then trained with the training set introduced before. Once the model is trained feeding the experimental data to the NN this can try to infer the values of the parameters required to get the output.

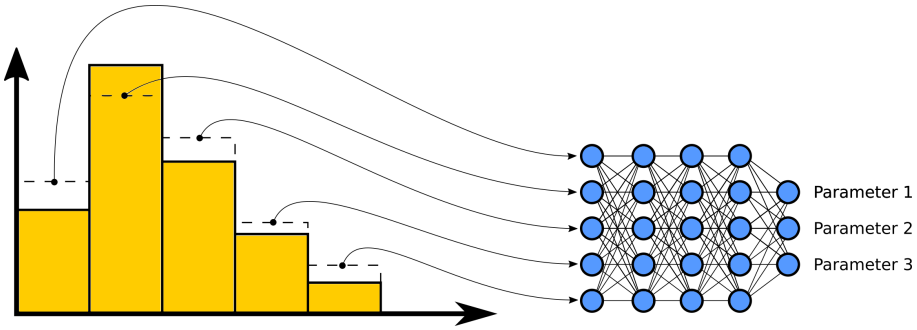


Figure 4.8: Figure from [7]

Errors evaluation

The errors are evaluated in a different way with respect to PerBin model. In fact, in the Inverse Model there is no a minimization step, and the error is evaluated by a re-sampling of the experimental data using a *multivariate Gaussian Distribution*, as the one in Eq. 4.11 with a diagonal covariance matrix that has experimental uncertainties on the main diagonal.

$$f(x_i; h^{(i)}_{\text{exp}}, \sigma^{(i)}_{\text{exp}}) = \mathcal{N} \cdot \exp\left[-\frac{1}{2}\sum_{j=1}^{N_{bins}} \frac{\left(x_i - h^{(i)}_{\text{exp}}\right)^2}{{\sigma^{(i)}_{\text{exp}}}^2}\right] \tag{4.11}$$

So, a set of histograms is generated, then this is fed to the NN and a distribution of predictions is generated. An example is shown in Fig. 4.9, from this distribution one can compute the error by evaluating the standard deviation.

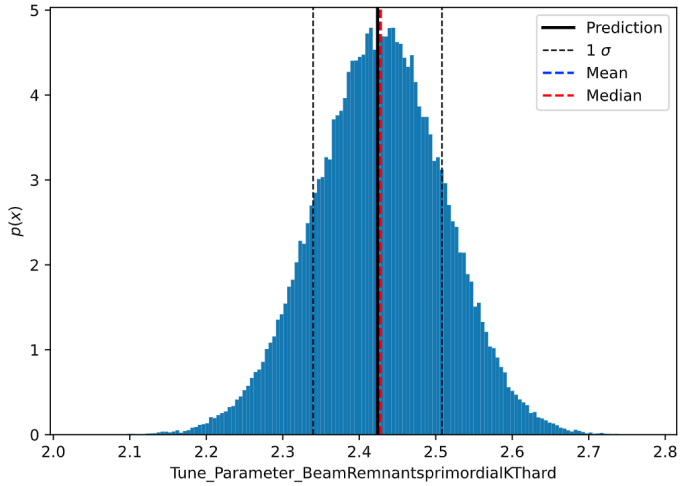


Figure 4.9: Predictions spread for the inverse model after that a Gaussian resample is performed on experimental data. The errors on the predicted parameter are compute by means of the standard deviation and are indicated by the dashed black lines.

Note that this is a new method for the tune. This method requires more attention than the PerBin Model to get it working correctly in the case of a high number of parameters to tune, this will be seen in the next chapter.
With respect to the PerBin Model this method is faster in the training step. The NN trained is only one and is not needed a minimization so a scan in the hyperparameters can be performed in order to search for the best architecture.

Hyperparameters

A really important step in the Inverse model is *hyperparameter optimization*. It is required to get the method working.

The procedure consists in building a *validation set* containing some Monte Carlo simulations as the training set (e.g. 10% of the simulations in the training set) and retrain the model with different choices for the NN architecture. Then a closure test is performed in order to estimate the performance of the NN. The test is performed using these validation set MC runs as input instead of the real experimental data, the closure test is done by computing a loss function between the predicted value and the Monte Carlo truth defined as:

$$L = \sum_i \frac{\left| p_i^{\text{true}} - p_i^{\text{pred}} \right|}{p_i^{\text{true}}} \tag{4.12}$$

The configuration that minimizes the distance between predicted and real values (MC truth), and so this loss function, is the best architecture in the selected hyperparameter space.
Once the best model is found, it is retrained using the MC runs contained in both the training and validation sets. Then the experimental data are fed to it in order to get the best estimation for the parameters to tune. This procedure is schematically summarized in Fig. 4.10.
The hyperparameters scan is performed using the python package `hyperopt`.

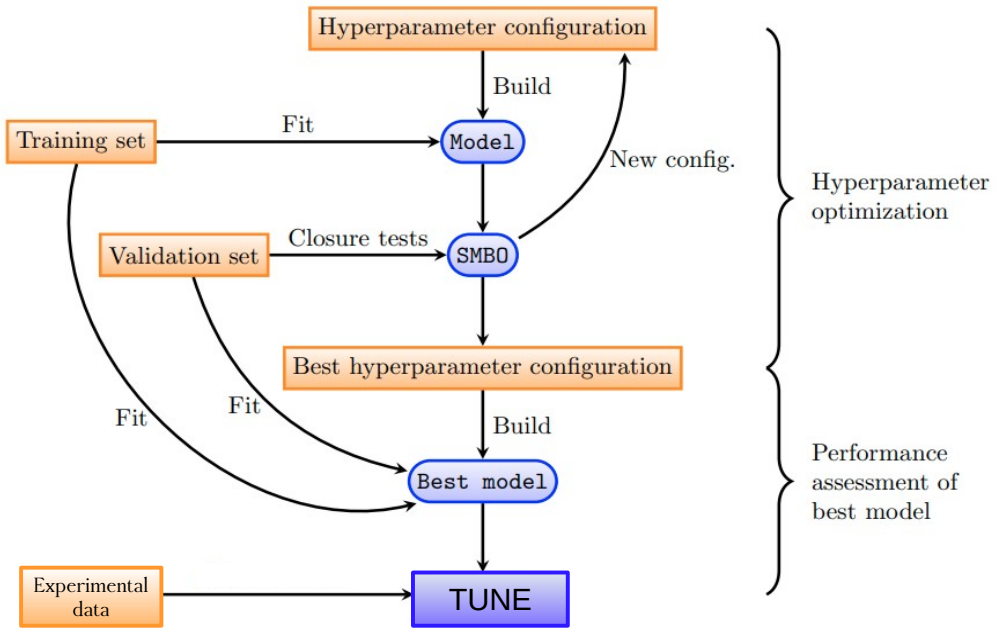


Figure 4.10: The hyperparameter optimization procedure is schematically shown here. The model is trained using a training set then performing a closure test a scan on the hyperparameter is done. Once the best configuration is found the best model is retrained using both the runs in the training set and the runs in the validation set. Then the experimental data are fed to the network and the best parameters are estimated. Figure from [7]

.

Problems

As shown in the next chapter the main problem for the Inverse model is the instability of the results in some cases. The operation that this model is trying to do is very hard the parameters can have some not trivial correlations. This complication can increase a lot the complexity in the operation of learning the inverse response of the generator. These difficulties in some cases lead to a bad prediction with parameters value predicted out of the imposed ranges and with errors in the order of the parameter value.
As it will be seen in the chapter that describes the new tune performed using the Inverse model in some cases when the number of parameters became larger and maybe the correlation between the parameters are important this model fails in the prediction.
But if more care is given to this model this can become a very powerful method for future tunes.

4.3.5 Weightrules in MCNNTUNES

MCNNTUNES as PROFESSOR implements the possibility of changing the weight of the singles bins in the distributions. This is a really useful feature in order to give some more importance to distributions and points that require a better description

by the tune. As it will be discussed later these weightrules can be used to get results more similar to CP5 in the MCNNTUNES tune proposed in this thesis. This feature gives the possibility to decide which distribution is more important in the tune. Increasing the weight of a bin this bin became more important in the overall χ^2 evaluation and it is better described by the simulations.

Chapter 5

Tune for the Underlying Event

This section focuses on the work done in order to reproduce a similar tune to CP5 for the underlying event and minimum bias observables. This is done to test the ability of MCNNTUNES of being a valid tool for the tuning of Monte Carlo generators with real data.
In order to validate MCNNTUNES has been decided to first perform a simpler tune with only two free parameters and then try to reproduce CP5 [6] with all the five parameters variation.

5.1 Introduction

A different tune has been performed for the underlying event and minimum bias observables using the same distributions, listed in Chapter 3.3.1 and used in CP5 tune. So, it was expected to get a similar result from the tune using MCNNTUNES.
Here is reported a quick reminder for the PYTHIA8 settings used in CP5 and in MCNNTUNE tunes:

- The NNLO PDF set used is NNPDF3.1 [55];
- an α_s value equal for all the processes set to 0.118 and running with a NLO evolution.
- The ISR is also ordered according to rapidity.

In the tuning procedure have been employed both the MCNNTUNES operation modes described in the previous chapter: PerBin and Inverse models.

5.2 First test: only two parameters variation

The first simplified test that was performed is the tuning varying only two parameters. The parameters chosen are the `MultipartonInteractions:pT0Ref` and `MultipartonInteractions:ecmPow`. These two parameters have been introduced in Section 2.5.1.
This simplified test aims is to check the correct operation of MCNNTUNES in a restricted parameters space.

Parameter Name	Value
`MultipartonInteractions:pT0Ref` [GeV]	$[1.0 - 3.0]$
`MultipartonInteractions:ecmPow`	$[0.0 - 0.3]$
`MultipartonInteractions:coreRadius`	0.7634
`MultipartonInteractions:coreFraction`	0.63
`ColorReconnection:range`	5.176

Table 5.1: Parameters space for the two parameters test. The other parameters are set to the CP5 default values.

Table 5.1 shows the parameters space used in this first case. The other parameters are set to the CP5 values (also these are reported in the table as a reminder).
During this first part, the two distributions related to the single diffractive and non-single diffractive event selection were not used for a bug in the routine of the analysis that has been fixed up before the real tune.
Now the results obtained for the test with the two models are discussed.

5.2.1 Per Bin Model results

The PerBin model estimation of the best parameters is performed by a loss function minimization. The output of the minimizer is reported in Fig. 5.1, the loss function (χ^2/dof) is the orange one, while the best parameter is marked by a solid red line.

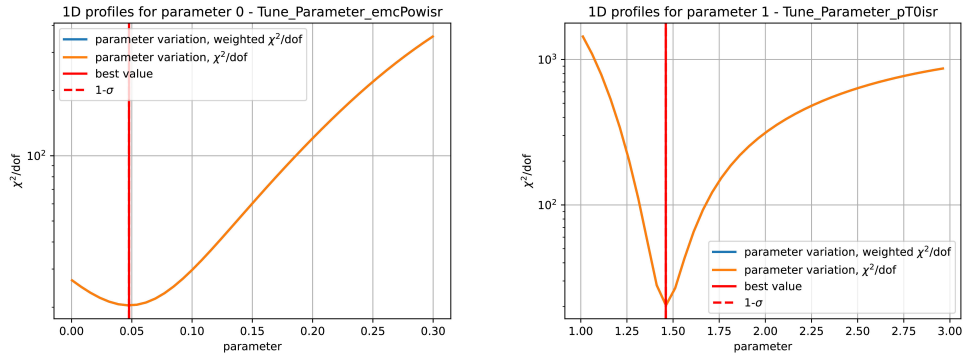


Figure 5.1: The figure shows the output of the minimizer for each parameter. The right one refers to the `MPI:ecmPow` while the left one to the `MPI:pT0Ref`. The orange line is the χ^2/DoF as a function of the parameter values, the solid red line indicated the best estimation for the parameters.

The estimated parameters for the tune are also reported in Table 5.4 with a comparison on CP5 boundary for each parameter. The errors on the parameters estimated with the PerBin model are not reported here because a correct error implementation, like the one described in Section 4.3.2, was not yet implemented in the software at the moment of this test, but here an example of the old error estimation can be seen. The errors estimated with the old method are too small to represent a credible interval, in fact, the error lines (dashed red lines) are overlapped with the line of the parameter estimation in both the figures. The predicted errors were 3 or 4 orders of magnitude smaller than the parameter values.

As was expected, the results derived from PerBin model are similar to the ones obtained from the CP5 tune. The distributions simulated by using these parameters are shown below in Section 5.2.3 together with results from the Inverse model and CP5.

An important observation is that the model of the generator response is more sensitive to some parameters with respect to others.
As an example, in Fig. 5.2 it is clear that the model is more sensitive to the parameter `MultipartonInteractions:pT0Ref` (top) respect to `MultipartonInteractions:ecmPow` (bottom). This is reflected in the output of the minimizer, in which a larger sensitivity corresponds to a better-defined minimum and vice versa. The top two panels show what happens when the distributions are very sensitive to the variation of the parameter as can be seen on the right side the variation of the `pT0Ref` parameter leads to very different scenarios. This is reflected by the minimizer out in a well-defined minimum and a smaller error on the parameter determination. A different scenario is displayed on the bottom panels of Fig. 5.2 where the distribution is less sensitive to the variation of the parameter `ecmPow` and so the minimum is less defined and so it will have a larger error on the estimation.

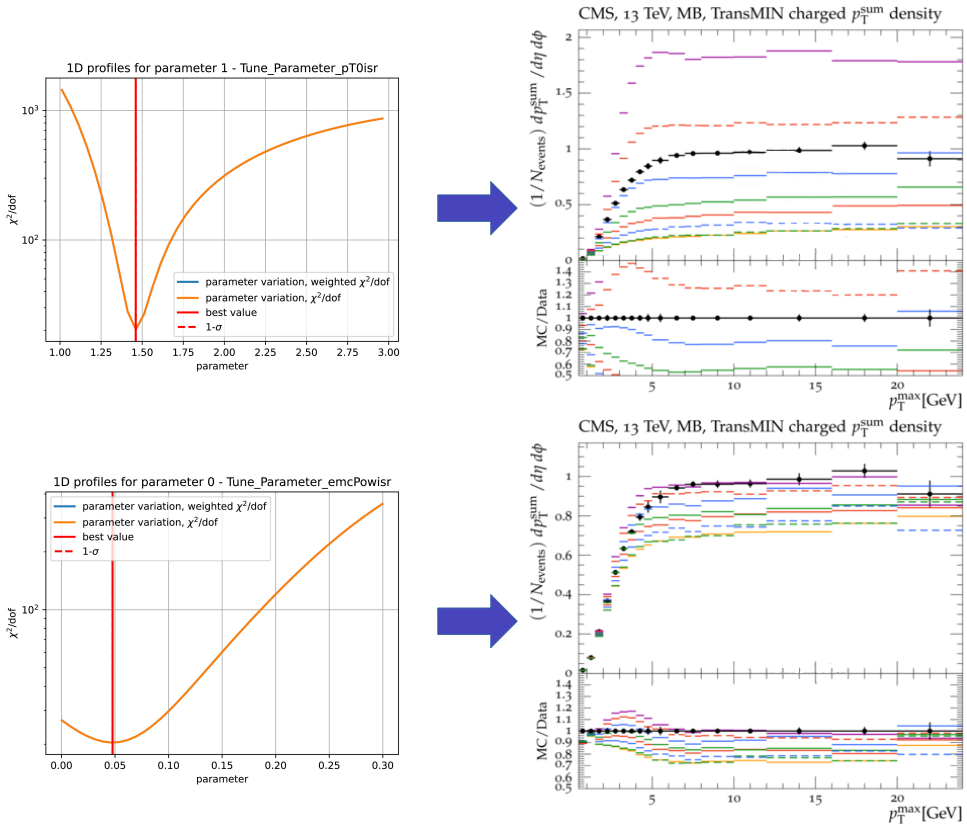


Figure 5.2: The sensibility of the tuned distributions with respect to the variation of the parameters related to the output of the minimizer. The top panels are related to `pT0Ref` while the bottom ones to `ecmPow`. The different colored lines in the 2 distributions (on the right) are the simulations with different values for the parameter. The black points are the data.

5.2.2 Inverse Model results

The other operation mode offered by MCNNTUNES is the Inverse model. The outcomes of the first test using the Inverse model are described here while the overall distributions are reported in the next section together with the ones from the PerBin Model.
As mentioned before to make this model work properly hyperparameter optimization is necessary. The hyperparameter optimization used here was a scan of the architecture parameters shown in Table 5.2, where the number of trials (combinations of these parameters) was 1000. In this case, the best model found is the one reported in Table 5.3.

Hyperparameter	Variation Range
Number of hidden layer	2-5
Units per layer	2-20
Activation function	tanh, relu, sigmoid
Optimizer	sgd, rmsprop, adagrad, adadelta, adam, adamax, nadam
Epochs	250-15000 in discrete steps
Batch size	64-5000 in discrete steps
Number of trials	1000

Table 5.2: Hyperparameter space scanned for the optimization of the NN architecture.

Hyperparameter	Value
Number of hidden layers	4
Units layers	$[2, 14, 9, 18]$
Activation function	tanh
Optimizer	nadam
Epochs	1000
Batch size	500

Table 5.3: Best hyperparameters model found for the test with 2 parameters variation. The "Units layers" row indicates the number of units in each layer.

Once the best model is trained the output obtained from this model is the distribution of predictions obtained by the resampling phase of the experimental data then fed to the network. The prediction spread for the two parameters test is shown in Fig. 5.3 these are obtained from the re-sampling phase using the multivariate Gaussian distribution described in Eq. 4.11.
The estimated parameter is marked by the solid black line, while the dotted black lines are the error on the parameter. The resulting values derived from the tune are also reported in Table 5.4 together with the CP5 values. In this case, the error estimation is correctly implemented and a direct comparison between the value is possible. Looking at the values reported in the table it is clear that the two tune are compatible.

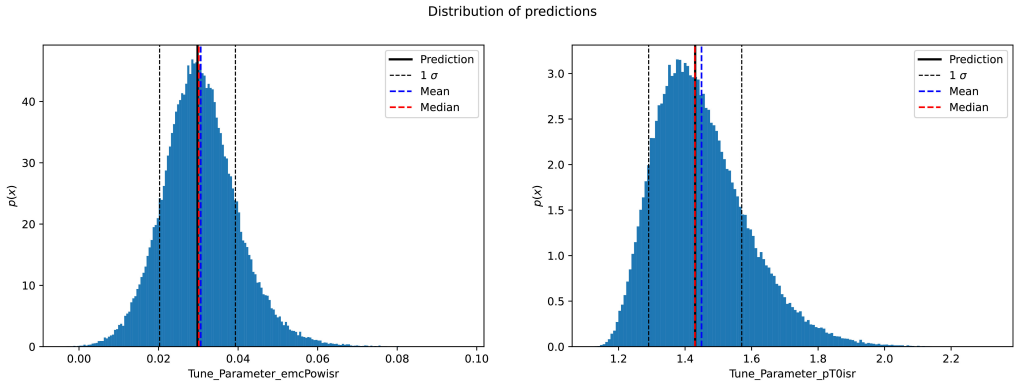


Figure 5.3: The spread of predictions that is produced as output from the Inverse Model. The right one refers to the `ecmPow` parameter and the left one to `pT0Ref`. The parameters predicted are marked by the solid black line while the error is evaluated using the standard deviation indicated by the dashed black lines.

5.2.3 Overall results

This section will show some overall results for the first test with only two free parameters. Not all the graphs are shown here, the other ones are listed in the Appendix A.

Parameter	PerBin	Inverse	CP5 (down & up)
`MPI:pT0Ref`	1.46064	1.43 ± 0.14	$1.41 - 1.46$
`MPI:ecmPow`	0.04771	0.0298 ± 0.0095	0.03

Table 5.4: Results for the PerBin and Inverse models in two-parameter variation test. The upper and lower limits for CP5 are also reported here for a direct comparison between the two tunes. The two predicted parameters for the Inverse model are compatible with the ones in CP5. A direct comparison is not possible for the PerBin model, due to the not properly evaluated error. But the estimated parameters are quite similar to the expected ones.

From the graphs in Fig. 5.4 is clear that MCNNTUNES gives a good agreement between the experimental data and the MC points.
The results obtained from the PerBin model (green line) and from the Inverse model (blue line) are similar to the result from the CP5 tune (red line). Tune employing MCNNTUNES describe the low region ($p_T \lesssim 5$ GeV) with a good agreement, this region is very important because is the region with lower error on the experimental data.

Overall, since the MC points are describing the experimental data well, the test can be considered successfully passed from MCNNTUNES. It gives results similar to the standard tool PROFESSOR. This was only a simplified test to check the correct operation of the tool. So, given the good result of the test phase, the next step is to extend the analysis to a complete tune for the underlying event trying to reproduce the CP5 tune.

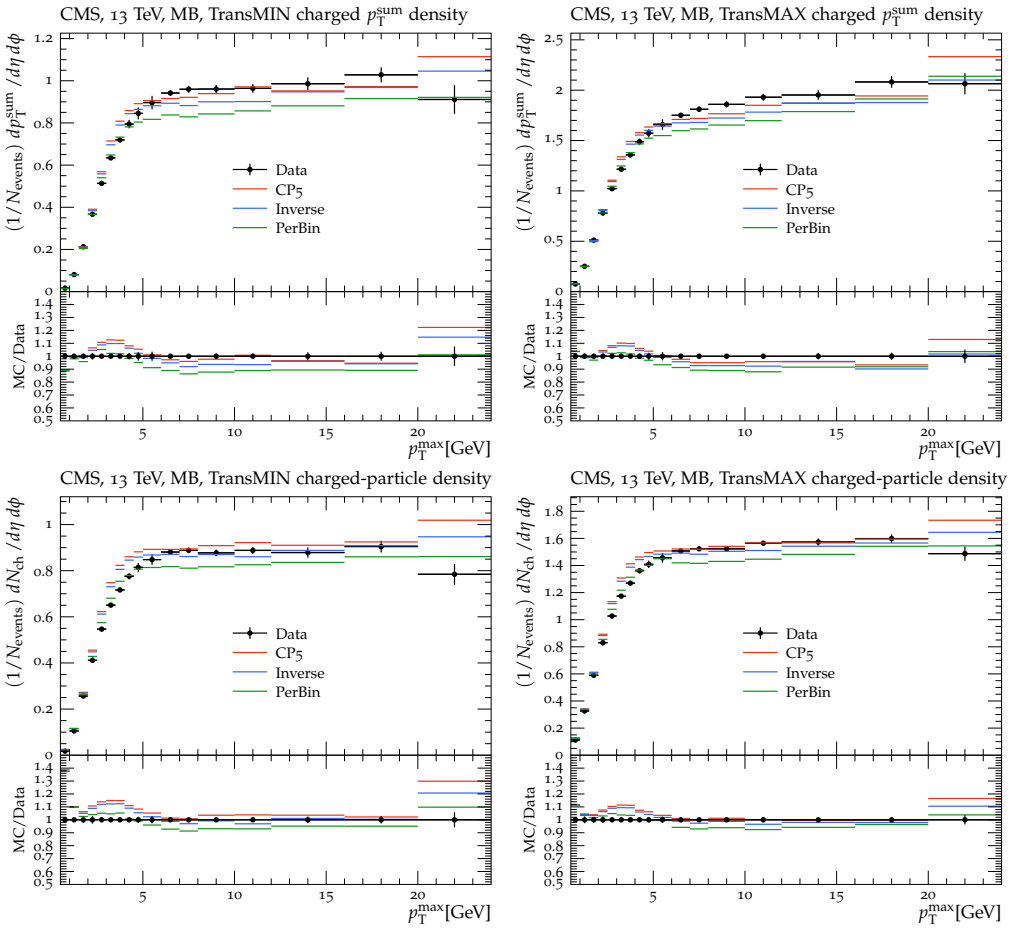


Figure 5.4: Results for the two parameters variation test. Here the distributions from the $\sqrt{s} = 13$ TeV CMS analysis [33] are reported. The shown observables are the transMAX charged particle density (upper left) and the charged p_T-sum density (upper right); the transMIN charged particle density (lower left) and the charged p_T-sum density (lower right) as a function of the transverse momentum of the leading charged particle. The black points are the experimental data and the black vertical lines the experimental uncertainties. The MC predictions are indicated by the colored lines. Also, a ratio plot between MC and data points is displayed below each distribution.

5.3 Tune for the Underlying Event

Given the good results of the test, the analysis has been extended to the variation of five parameters. The interesting parameters are the ones related to the Multi Parton Interaction and to the Color Reconnection. The ranges of variation for these parameters are the same used for CP5 and summarized in Table 5.5.

Parameter Name	Value
`MultipartonInteractions:pT0Ref` [GeV]	$[1.0 - 3.0]$
`MultipartonInteractions:ecmPow`	$[0.0 - 0.3]$
`MultipartonInteractions:coreRadius`	$[0.1 - 0.95]$
`MultipartonInteractions:coreFraction`	$[0.1 - 0.8]$
`ColorReconnection:range`	$[1.0 - 9.0]$

Table 5.5: The variation ranges for the five parameters that will be tuned using MCNNTUNES. These are the same used in the CP5 tune in [6].

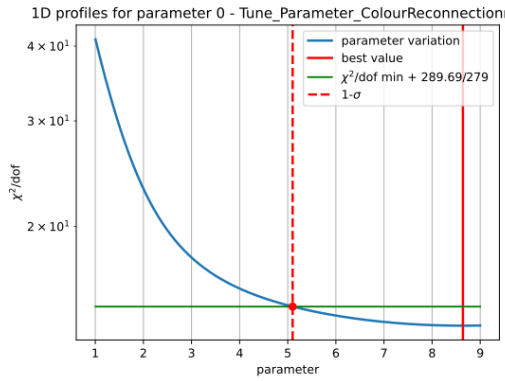
As the parameters space increase, one needs to increase also the number of samples and then the size of the training set in order to have sufficient granularity in the sampling of the space. The training set used here for the PerBin model with five parameters variation is composed of approximately 2000 MC runs.

5.3.1 Per Bin Model results

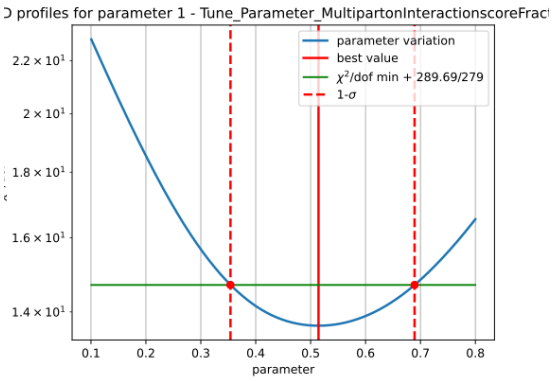
The parameters estimation achieved from the PerBin model loss function minimization is reported in Fig. 5.5. In the figure, the five parameters χ^2/DoF functions are reported with a blue line. The predicted value is indicated by the solid red line while the $1 - \sigma$ range with the dotted lines. It is clear that also in this case the most sensitive parameter is the `MultipartonInteractions:pT0Ref` (5.5e) the minimum, in that case, is very well defined and the error small. The parameters in Fig. 5.5c and Fig. 5.5d are also well defined the error is not to big. The worse defined parameters are the `MultipartonInteractions:coreFraction`, which is in a minimum but with a larger error than the other parameters which is why the distributions under analysis are less sensitive to the variation of this parameter, and the `ColorReconnection:range`. The last one is not actually in a real minimum, in this case also the error evaluation, as a confidence interval is not possible. This could indicate that above a certain value the MC distributions are no longer sensitive to the variation of this parameter. In fact, over a certain value all the possible reconnections have already occurred, and taking a larger value of this parameter does not improve nor worsen the description of the data distribution by means of the MC simulation.

The obtained parameter values are reported also in Table 5.6 and compared to the CP5 limits. It is easy to see that all the parameters are compatible with the CP5 except for the `ColorReconnection:range`.

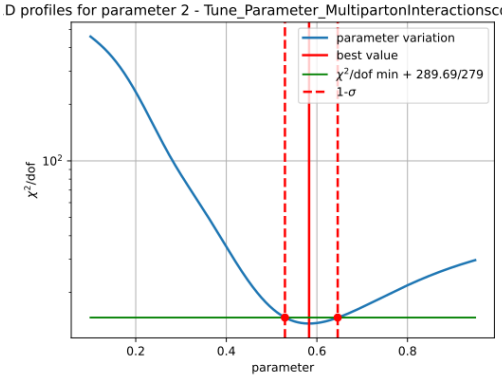
The distributions obtained are reported in the Section 5.3.3 as can be seen in the Fig. 5.13 the PerBin model does not describe very well this distribution for the pseudorapidity of the inelastic production of hadrons. There is a high discrepancy



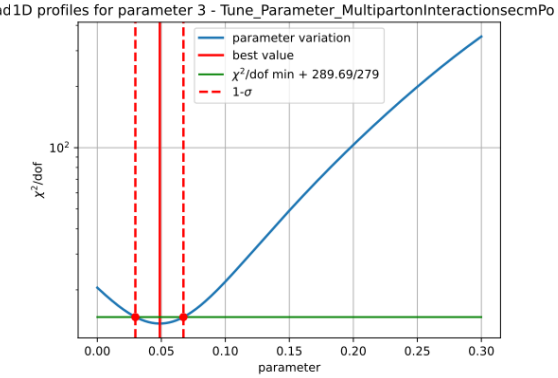
(a) `ColorReconnection:range`



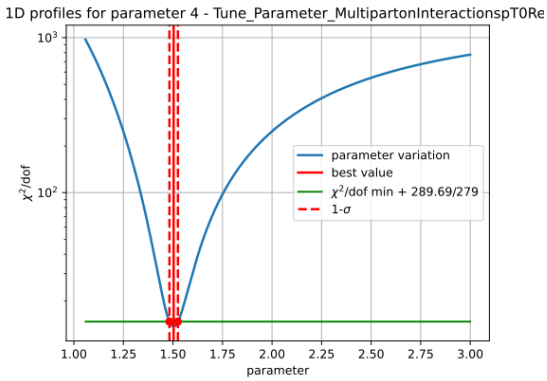
(b) `MultipartonInteractions:coreFraction`



(c) `MultipartonInteractions:coreRadius`



(d) `MultipartonInteractions:ecmPow`



(e) `MultipartonInteractions:pT0Ref`

Figure 5.5: The minimizer output for every tuned parameter. The blue line is the $\chi 2/DoF$ as a function of the parameter value. The solid vertical line indicates the best estimation for the parameter value while the errors are indicated by the dashed red lines. In the graph (a) it is clear that over a certain value of this parameter the description of the data by the MC simulation with respect to this parameter variation has a small sensitivity, and this leads to a non-well defined minimum. The upper uncertainty, in this case, cannot be calculated. The other parameters are all in a real minimum and with a reliable error evaluation.

between MC and data points in this distribution. A possible explanation is that the experimental uncertainties on the bins of this distribution are larger and so this

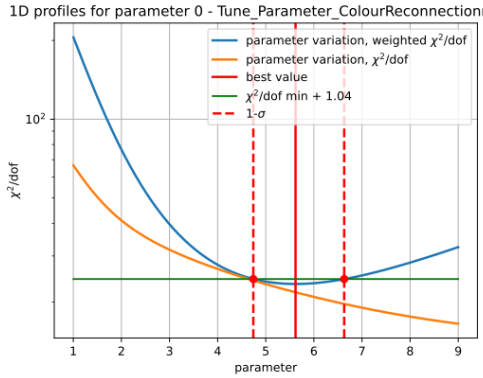
Parameter	Value	CP5 (down & up)
`MultipartonInteractions:pT0Ref`	$1.50^{+0.02}_{-0.02}$	$1.41 - 1.46$
`MultipartonInteractions:ecmPow`	$0.049^{+0.018}_{-0.019}$	0.03
`MultipartonInteractions:coreFraction`	$0.51^{+0.17}_{-0.16}$	$0.43 - 0.73$
`MultipartonInteractions:coreRadius`	$0.58^{+0.06}_{-0.05}$	$0.67 - 0.69$
`ColorReconnection:range`	$8.6^{-3.5}_{+null}$	$4.88 - 4.69$

Table 5.6: The results of the tune using the PerBin Model. All the values are compatible with respect to those of CP5 using a Z test with a significance level of 0.05. But the value predicted for the Color Reconnection has a very large error and is not as similar to the one obtained from CP5. The *null* subscript indicates that the definition of the confidence region exceeds the scanned space for the parameters defined for each parameter in Table 5.5.

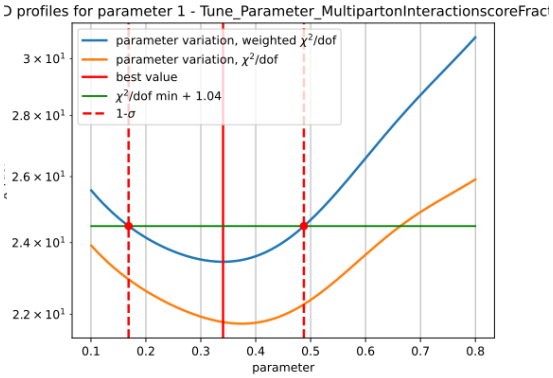
distribution is less important in the overall loss function.
So, another tune using PerBin Model has been performed. In this second tune, it has been given to all Fig. 5.13 bins a higher weight using the weightrules implemented in MCNNTUNES, a weight of 5 in the minimization function has been given to all bins of this distribution. In this way, greater importance is given to this distribution and so the MC simulation will try to describe these data better as a consequence of the larger importance in the final loss function for these bins.
The output of the mimization step is reported in Fig. 5.6. The parameter values obtained from the weighted tune are also reported in Table 5.7. The values of the parameters are compatible also in this case with CP5 except for the `MultipartonInteractions:coreRadius` that is higher than the one predicted from CP5.
But if one looks at the overall result in Section 5.3.3 the PerBin Model with different weights gives a result more similar to CP5 in almost all the distributions.

Parameter	PerBin	PerBin + Re-weight	CP5 (down & up)
`MPI:pT0Ref`	$1.50^{+0.02}_{-0.02}$	$1.42^{+0.01}_{-0.01}$	$1.41 - 1.46$
`MPI:ecmPow`	$0.049^{+0.018}_{-0.019}$	$0.0342^{+0.014}_{-0.014}$	0.03
`MPI:coreFraction`	$0.51^{+0.17}_{-0.16}$	$0.34^{+0.14}_{-0.17}$	$0.43 - 0.73$
`MPI:coreRadius`	$0.58^{+0.06}_{-0.05}$	$0.9^{+null}_{-0.097}$	$0.67 - 0.69$
`CR:range`	$8.6^{-3.5}_{+null}$	$5.6^{+1.0}_{-0.9}$	$4.88 - 4.69$

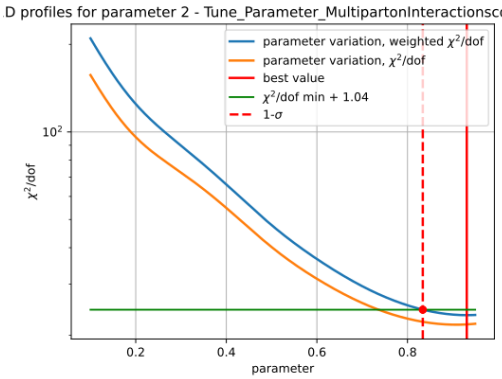
Table 5.7: The results of the tune using PerBin model + re-weight are compared to the ones with PerBin model and CP5. except for the coreRadius parameter, all the predicted values are compatible with the one with the CP5 using a Z test with a significance level of 0.05. The PerBin + re-weight gives results more similar to the one obtained from CP5 this is also clear watching to the overall resulting distribution. The *null* subscript indicates that the definition of the confidence region exceeds the scanned space for the parameters defined for each parameter in Table 5.5.



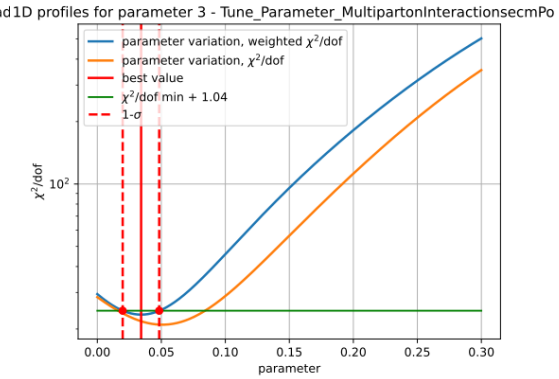
(a) `ColorReconnection:range`



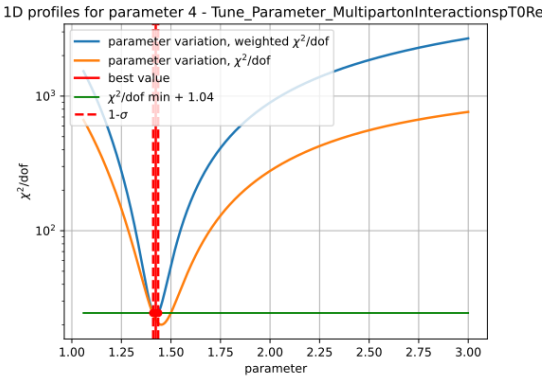
(b) `MultipartonInteractions:coreFraction`



(c) `MultipartonInteractions:coreRadius`



(d) `MultipartonInteractions:ecmPow`



(e) `MultipartonInteractions:pT0Ref`

Figure 5.6: The minimizer output for every tuned parameter using PerBin model with the re-weight. The blue line is the $\chi 2/DoF$ as a function of the parameter value before the re-weight while the orange line indicates the same function but applying the re-weight. The solid vertical line indicates the best estimation for the parameter value while the errors are indicated by the dashed red lines. The parameter in figure (c) is predicted near the upper limit of the variation range.

5.3.2 Inverse Model results

In the case of five free parameters, the performance of the Inverse model is not as good as the one of the PerBin model. In this case, the Inverse Model gives a bad description of the data the MC points are far from the experimental data. Also after the hyperparameters optimization, using the scan space described in Table 5.2, the model fails. The output of this hyperparameter optimization is reported in Table 5.8.

Hyperparameter	Value
Number of hidden layers	4
Units layers	[8, 18, 34, 16]
Activation function	tanh
Optimizer	adamax
Epochs	11000
Batch size	5000

Table 5.8: Best hyperparameters model found for the test with 2 parameters variation. The "Units layers" row indicates the number of units in each layer.

The distributions of predictions are reported for each parameter in Fig. 5.7 and the actual parameters in Table 5.9 as one can see the predicted values are in most of the cases out of the variation ranges set for the sampling and (also to the maximum possible value in PYTHIA8). Some other parameters are determined with a very large distribution and so very large errors.

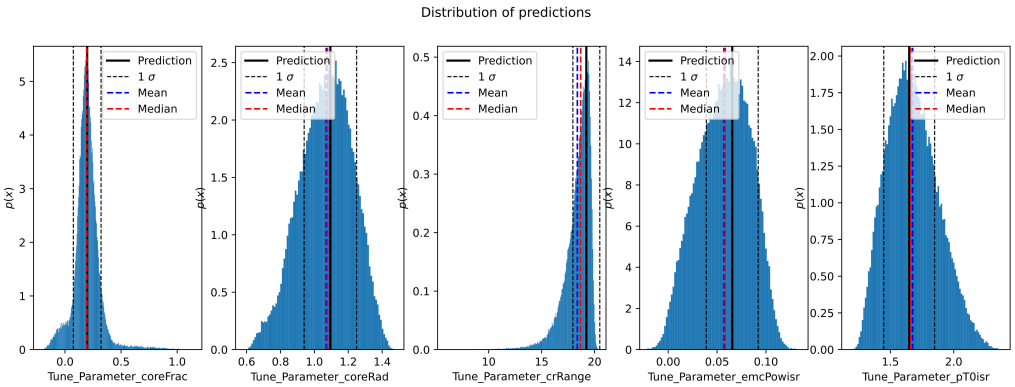


Figure 5.7: The spread of prediction that is obtained from the Inverse model. It is clear that also after the hyperparameters optimization the inverse model is not working properly. The distributions have long tails out of the limits for the variation, the central one referred to the Color Reconnection is predicted completely out of the boundaries.

It is easy to see from Fig. 5.8 that the tune obtained from the Inverse Model (blue line), as was expected, cannot describe the distributions of the observables. The tune misses all the experimental data points approximately by 50% of the value. Se this model has been excluded for this tune. Maybe the failure is related to the higher number of parameters respect to the first test, that leads to a more complex generator response and so a more difficult model to learn and invert.

Parameter	Value	CP5 (down & up)
`MultipartonInteractions:pT0Ref`	1.65 ± 0.20	$1.41 - 1.46$
`MultipartonInteractions:ecmPow`	0.066 ± 0.026	0.03
`MultipartonInteractions:coreFraction`	0.20 ± 0.12	$0.43 - 0.73$
`MultipartonInteractions:coreRadius`	1.1 ± 0.2	$0.67 - 0.69$
`ColorReconnection:range`	19.2 ± 1.3	$4.88 - 4.69$

Table 5.9: The predicted values using Inverse model are shown here, the model is not working properly in this case, the predicted values for the core radius and for the color reconnection are out of the boundary while other parameters are predicted with a quite large error.

The MCNNTUNES paper [7] reports that the Inverse model can fail when in the training set is not given a sufficient number of MC simulations with parameters values near to the actual real ones. So, trying to make this model work, it has been decided to introduce a Gaussian bias in the sampling phase instead of a uniformed distributed sampling. The Gaussian sampling was peaked on the parameters predicted by the PerBin Model, but also this procedure does not lead to consistent results.

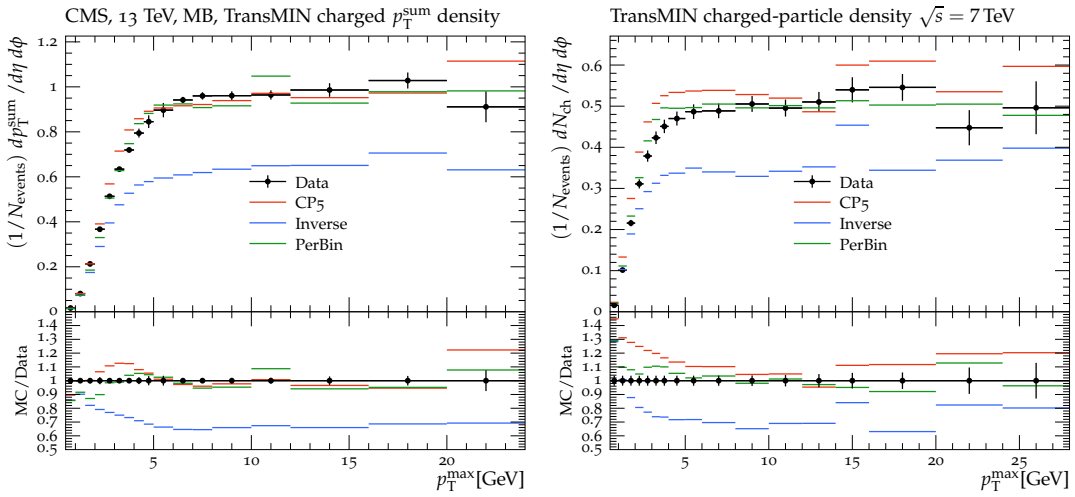


Figure 5.8: An example of the fact that the Inverse model (blue line) is not working: the bins are filled to only the 50% of the expected value in almost all the bins. On the left is shown the transMIN charged particle transverse momentum sum density for the CMS analysis at $\sqrt{s} = 13$ TeV [33] and on the right the transMIN charged particle density from the CMS analysis [34] in both cases as a function of the leading charged particle/charged particle jet transverse momentum.

5.3.3 Overall results

In conclusion, two good tunes have been obtained using the MCNNTUNES PerBin Model In this section, all the fitted distributions are shown and the two tunes are compared to the distribution obtained with the CP5 default values. The PerBin Model describe very well the data in particular in all the low-p_T regions where the experimental uncertainties are smaller and so more important in the χ^2 evaluation.

The first distributions shown in Fig. 5.9 are related to the charged particle multiplicity and the charged particle scalar p_T sum in the two transverse regions (TransMAX and TransMIN) as a function of the leading object transverse momentum. The black points are the experimental data taken from the CMS experiment at the center of mass energy of 13 TeV. They are compared to the CP5 tune, red line, and the two tunes from the PerBin Model and PerBin Model plus the re-weights, respectively blue and green lines. It is easy to see that all the three tunes are describing the distribution very well, in particular, tunes based on MCNNTUNES in the low-p_T regions ($p_T < 5$ GeV) are describing the distribution also better than CP5 in most of the cases.
Instead, in Fig. 5.10 are reported the same distribution but at $\sqrt{s} = 7$ TeV also in this case tunes based on MCNNTUNES are good in describing the distributions, they seam to be also better than CP5.
Also the data at the center-of-mass energy of 1.96 TeV are well described. These data had been collected by the CDF experiment in proton-antiproton collisions.
Fig. 5.12 and Fig. 5.13 represent the pseudorapidity distribution for diffractive events and for inelstaic charged hadrons production. The left distribution of Fig. 5.12 and the distribution in Fig. 5.13 are not well described by the PerBin Model (blue line) but are better described by the PerBin Model plus the re-weights for these bins. So also in this case a good result has been obtained from MCNNTUNES tool.

A numerical evaluation of the overall difference between MC and experimental points can be derived for all the three tunes using the χ^2 definition:

$$\chi^2 = \sum_i \frac{(\mathrm{MC}_i - \exp_i)^2}{\sigma_i^2} \quad . \tag{5.1}$$

The obtained results are reported in Table 5.10. From this is clear that from this

Tune	χ^2/DoF
CP5:	23.9
PerBin:	13.7
PerBin + re-weight:	19.4

Table 5.10: χ^2 evaluation for the three tunes.

evaluation the better tune overall is the PerBin model tune. In fact, the PerBin Model describes very well the distributions were the experimental uncertainties are smaller but don't describe very well the last distributions in the left panel of Fig. 5.12 and Fig. 5.13. A very good result is also the one obtained from the PerBin Model with the different weights for the bins in the distribution in Fig. 5.13. This tune, in the idea of having a more general tune that describes better all the distributions, can be considered a better tune. In general, the χ^2 evaluated is less in both MCNNTUNES-based tunes respect to the CP5 one.
Given these results, it is possible to conclude that MCNNTUNES is a valid tool for the tune of the parameters in high energy physics generators. Two valid results have been obtained for the tune of the underlying event in proton-proton collisions using the PerBin model and the PerBin model plus re-weights.

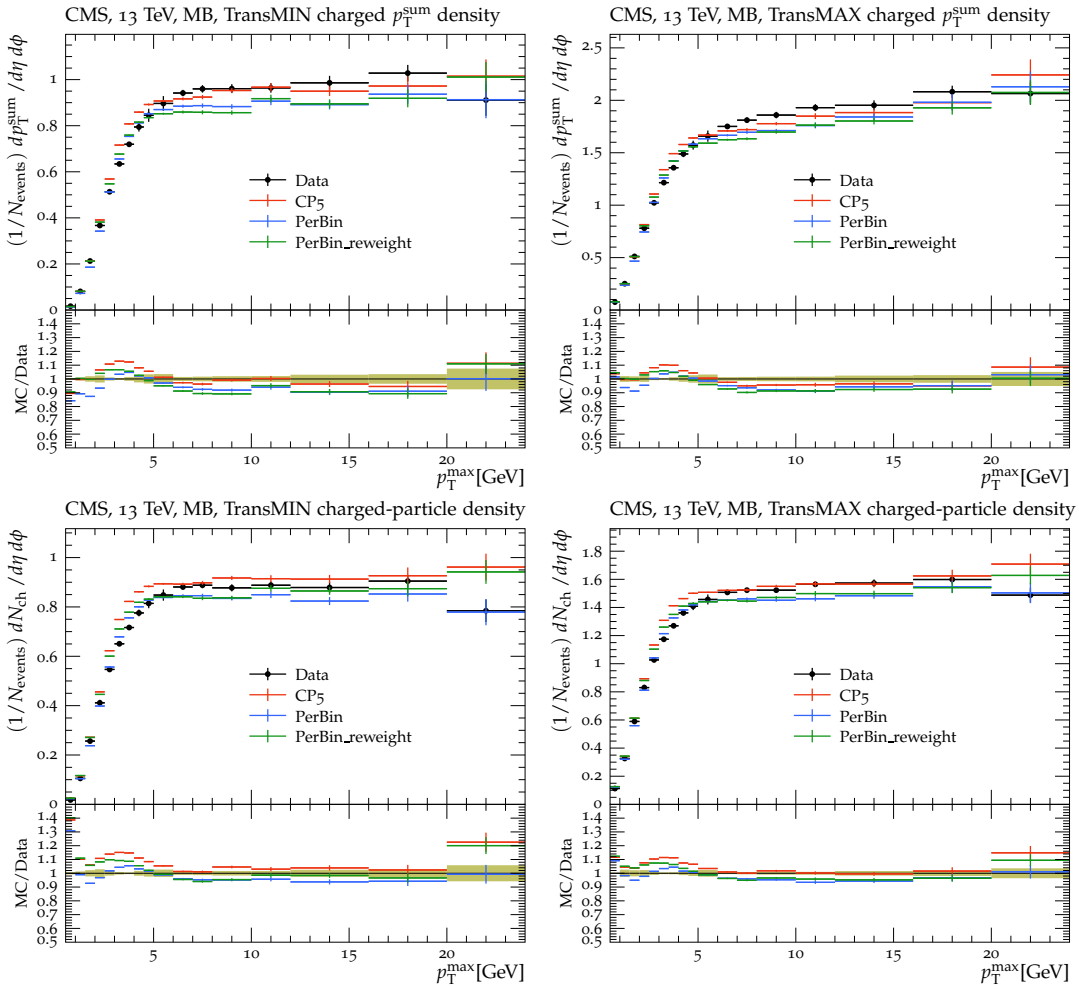


Figure 5.9: In this figure the data from the $\sqrt{s} = 13$ TeV CMS analysis [33] are displayed. The distribution showed are transMAX charged particle density (upper left) and the charged p_T-sum density (upper right); the transMIN charged particle density (lower left) and the charged p_T-sum density (lower right) as a function of the transverse momentum of the leading charged particle. The CP5 tune is compared to MCNNTUNES tune using the PerBin Model. PerBin model tune (red line) seems good as the CP5 (blue line) in describing the data. The first bins are the most important they have a smaller experimental error than the higher p_T data. Also the PerBin model with re-weight (green line) seems really good in the description of the data. Also the ratio between MC and data points is reported and the green band represent the experimental uncertainties, while the vertical colored lines on the MC points are the statistical uncertainties.

The activity observed in the two transverse regions and the pseudorapidity distributions for different event selections are well described after the tune of the parameters that control the MPI and the CR in PYTHIA8.
The outcome of the minimizer in Fig. 5.5 indicates that the most important parameter for the description of these distributions is the threshold value p_{T0}^{ref} of the MPI. All the distributions have a large sensitivity to the variation of this parameter and this is related to the well-defined minimum obtained.
On the other hand the small dependence on the variation of the `ColorReconnection:-`

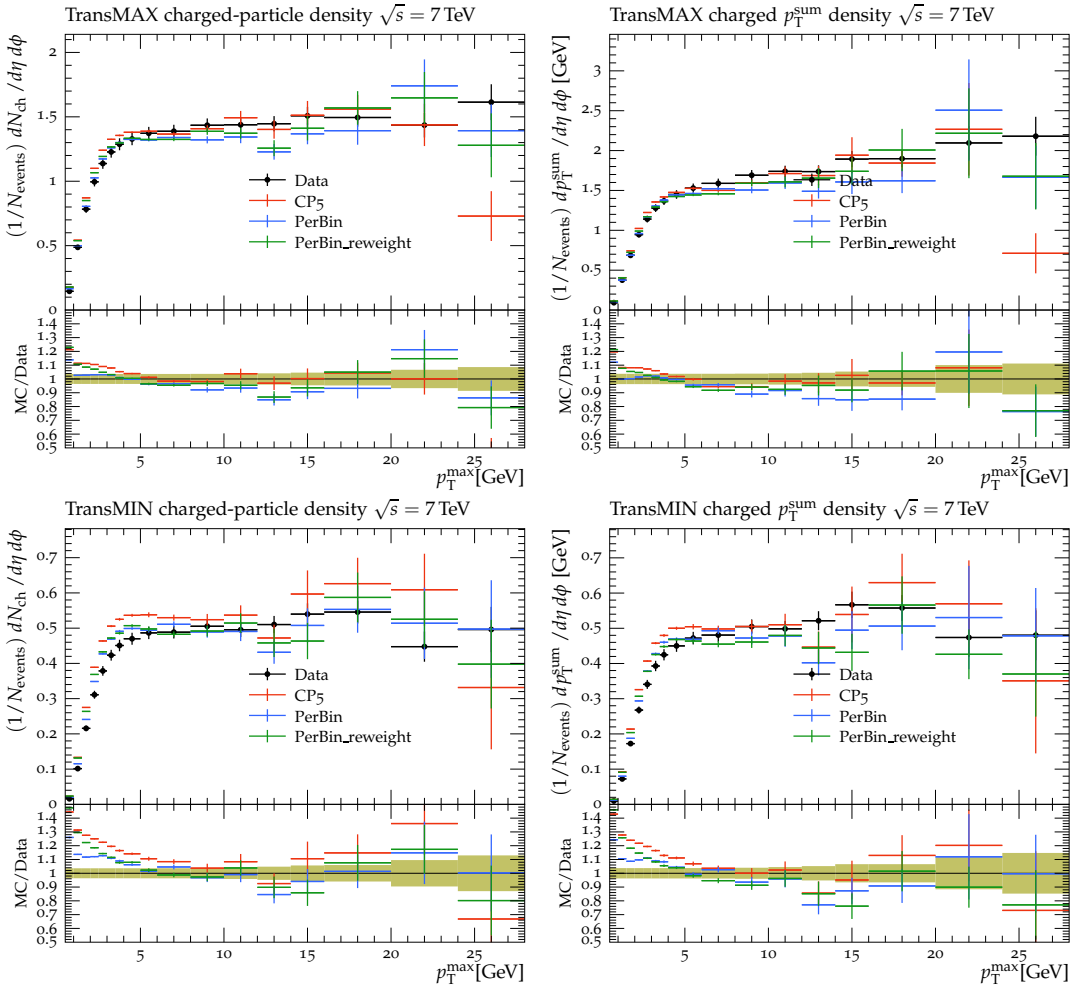


Figure 5.10: Here the data at $\sqrt{s} = 7$ TeV from the CMS analysis [34] for the transMAX charged particle density (upper left) and the charged p_T-sum (upper left); the transMIN charged particle density (lower left) and the charged $p - T$-sum are displayed as a function of the leading object transverse momentum. The three tunes describe the data (black point) very well. The low $p - T$ regions are described better from tunes based on MCNNTUNES respect to the CP5 tune. As was expected the results from the PerBin model with the re-weights (green line) are more similar to the CP5 result. Also the ratio between MC and experimental points is reported and the green band represent the experimental uncertainties, while the vertical colored lines on the MC points are the statistical uncertainties.

`range` parameter above a certain threshold value indicates that above this threshold all possible color reconnections have occurred and that therefore the model is no longer very sensitive to this parameter.
The Inverse model instead in this case does not give us a good result but maybe further tests could improve its performance. Maybe increasing, even more, the training set size but careful, MCNNTUNES does not present any control on the NNs typical over-fitting problem. Even if the Inverse Model does not give us a complete tune for the underlying events it performs very well in the first test. So, one cannot exclude it as a valid tool.
Now, that the tool is validated, the tuning procedure can be extended to some new

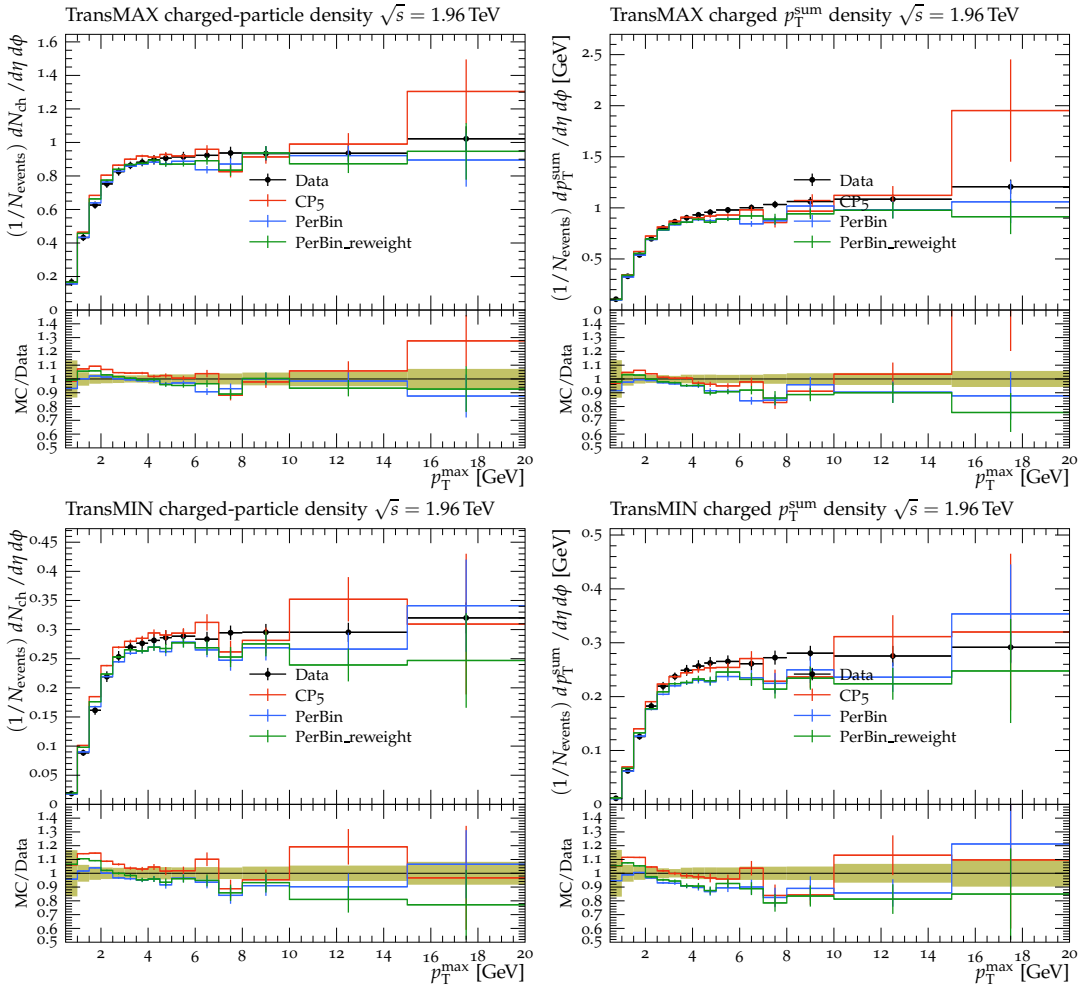


Figure 5.11: The transMAX charged particle density (upper left) and the charged p_T-sum (upper left); the transMIN charged particle density (lower left) and the charged p_T-sum from the CDF analysis at $\sqrt{s} = 1.96$ TeV in proton-antiproton collisions [35]. The data are described very well from the CP5, PerBin, PerBin + re-weights tunes. Also the ratio between MC and experimental points is reported and the green band represent the experimental uncertainties, while the vertical colored lines on the MC points are the statistical uncertainties.

distributions and parameters. The next chapter will focus on the tune of the *Primordial* k_T and Space Shower in order to improve the description of the data collected in Z boson production events.

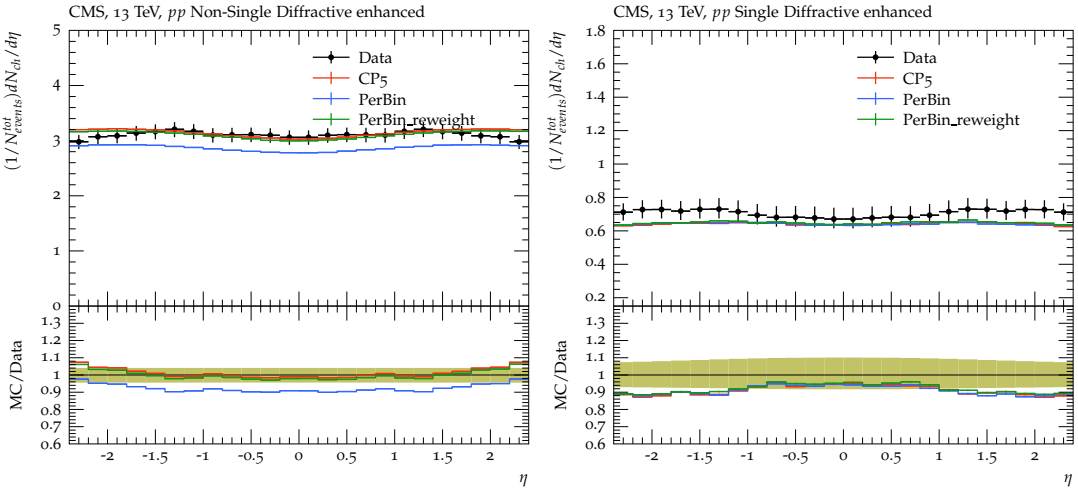


Figure 5.12: Here, the pseudorapidity distributions ($p_T > 0.5$ GeV, $|\eta| < 2.4$) for charged-particle multiplicity in single diffractive (right) and non-single diffractive (right) events selection are reported. The black point are the data from the CMS analysis at $\sqrt{s} = 13$ TeV [39]. The data from the NSD events are not so well described from the PerBin model (blue line) but with the re-weights, it is possible to describe these data points better. Instead, for the SD events a result equal to the one of CP5 has been obtained. Also the ratio between MC and data points is reported and the green band represent the experimental uncertainties, while the vertical colored lines on the MC points are the statistical uncertainties, in these distributions the statistical uncertainties are very small due to the high number of events in each bin.

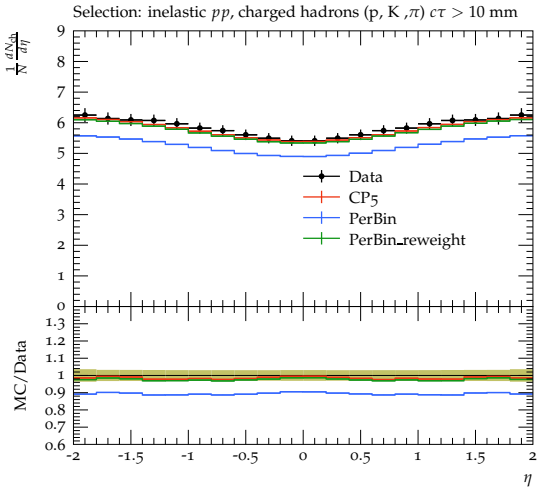


Figure 5.13: In this figure is shown the last distribution used for the tune from the CMS analysis at $\sqrt{s} = 13$ TeV [40]. The pseudorapidity distribution ($|\eta| < 2$) for the charged hadron density in an inelastic proton-proton scattering selection. Also here the PerBin model (blue line) gives a different result from the CP5 and the PerBin + re-weights tunes. Also the ratio between MC and data points is reported and the green band represent the experimental uncertainties, while the vertical colored lines on the MC points are the statistical uncertainties. In this distributions, the statistical uncertainties are very small due to the high number of events in each bin.

Chapter 6

Tune for the primordial k_T events

The primordial k_T is another important parameter in the description of the proton-proton collisions with Monte Carlo simulations. The primordial k_T description in PYTHIA8 was introduced in Section 2.6.
The main unresolved problem for the primordial k_T tune is the unexpectedly high value required for the description of the observed Z boson p_T spectrum.
In fact, the primordial k_T is derived by the Fermi motion of partons inside the hadrons. So when the parton undergoes the hard scattering it can already have an initial non-zero transverse momentum. The value of the primordial k_T can be estimated as reported in Eq. 2.45, but experimental data for the Z boson p_T spectrum show that this estimation is not sufficient. The required value estimated in order to reproduce the experimental data is of the order of 2 GeV.
The primordial k_T was set by the Monash 2013 tune [47] and the tunes derived from it, as CP5, inherited the value of this parameter.
So the introduction of a new tune for the primordial k_T is needed. In fact the CP5 tune is known not to describe well the Z spectrum in the low-p_T region [6]. This is shown in Fig. 6.1 for the production cross-section in Z plus jets events and in the figure Fig. 6.2 for Z boson production in DY observations. The CP5 tune misses the description of the experimental data values in the low regions of these observables. The distribution for Z+jets events as a function of the variable $p_T^{bal} = \left|p_T^Z - \sum_{\text{jets}} p_T^j\right|$ is not well described in the first bin. While the p_T^Z distribution is not well described in the region $p_T^Z \lesssim 40$ GeV. These two observables as described in [6] are very sensitive to the parton shower process and so to the UE.
The MCNNTUNES-based tune, performed here, focuses on the distributions in Fig. 6.2 where the effects of MPI are expected to be less important[1].

6.1 Primordial k_T and ISR effect on p_T^Z

The parameters that have been investigated in order to tune data and explain the Z boson p_T spectrum are:

- `BeamRemnants:primordialKThard` that set the width of the Gaussian distri-

[1] below it is preliminarily investigate the effect of the MPI on these distributions. This aspect is never been investigated in detail and requires further studies maybe also the inclusion of data from LHC RUN3 can improve the description of these observables.

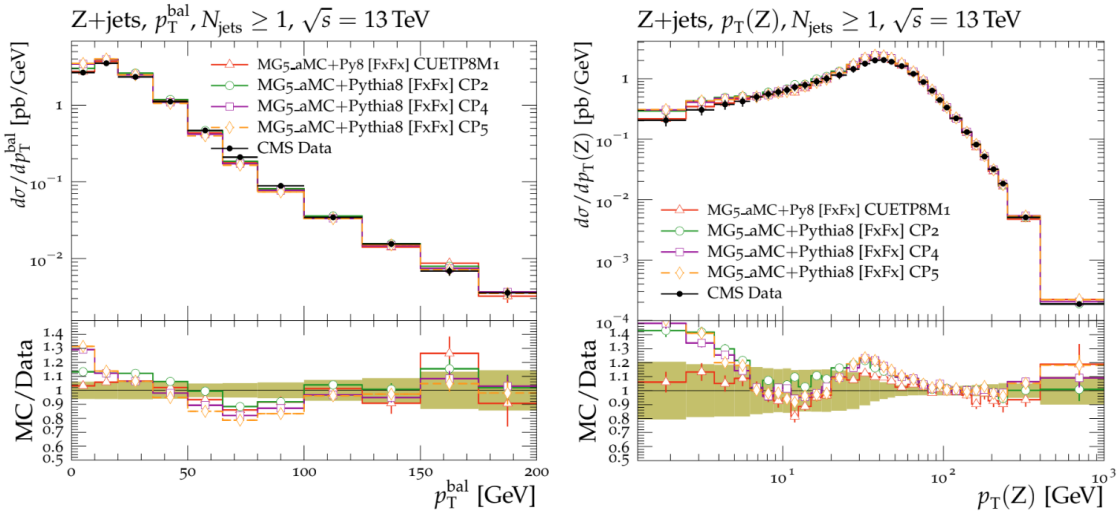


Figure 6.1: Figure taken from CP5 tune paper [6]. These two images show the Z boson production cross-section in Z plus jets (with at least one jet) at $\sqrt{s} = 13$ TeV as a function of the imbalance of the transverse momentum between the jet and the Z boson (left) and of the Z boson transverse momentum (right) [56]. The first bin in the left distribution and the region $p_T^Z \lesssim 40$ GeV, in the right one, is not well described by the CP5 tune (yellow line).

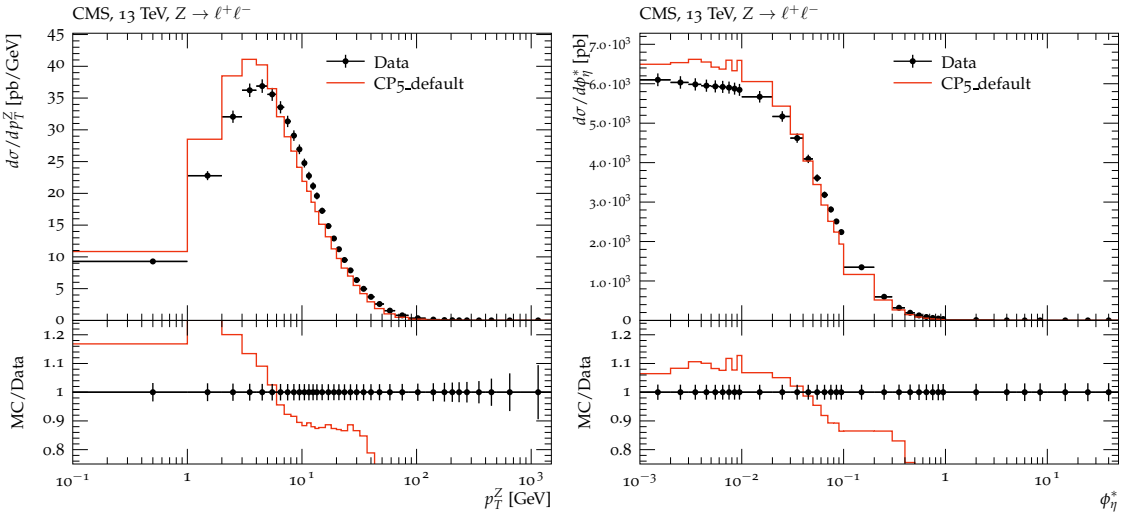


Figure 6.2: The CP5 tune (red line) compared to the Z boson production cross-section as a function of the Z boson transverse momentum (left) or of the ϕ_η^* angle. This distributions showed here are from the CMS analysis [46] at the center-of-mass energy of 13 TeV. The high region of the spectrum is not well described because it has been simulated only event to the LO, while a good description of it requires higher order matrix calculations.

bution for the primordial k_T sampling;

- `SpaceShower:pT0Ref` that sets the threshold for the initial state radiation to take place.

Now, it is discussed how this parameter impact the Z boson production transverse momentum spectrum. If only the LO diagram for the Z production (Fig. 6.3a) is

considered, one can only have a Z production with zero transverse momentum. So the Z boson p_T^Z spectrum expected at LO is a δ-distribution function centered on zero.

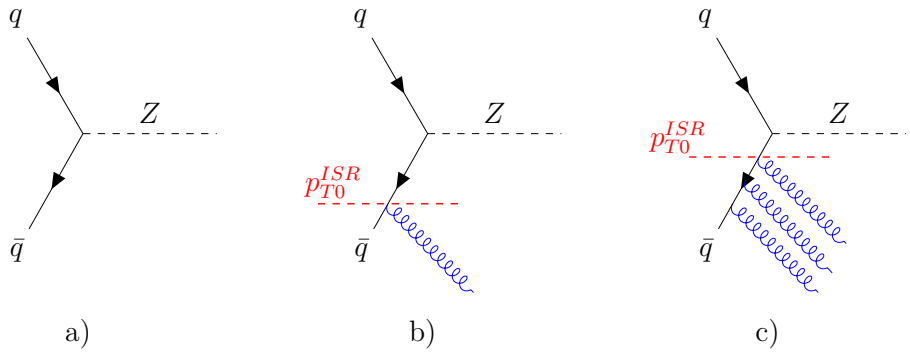


Figure 6.3: Z boson production diagrams: a) shows the LO diagram; b) is the LO diagram whit a high threshold for the ISR so it is produced a small amount of it; c) is the same but with a lower threshold for the ISR and a larger amount, in b) and c) cases the Z boson can be produced with a non-zero transverse momentum, balanced by the ISR.

With the addition of the primordial k_T a Gaussian distribution is obtained, whose width is set by `BeamRemnants:primordialKThard`. If one moves to higher-order diagrams the Z boson can be produced with a non zero p_T that is balanced by the various jets, then the primordial k_T is added to the already non-zero Z boson transverse momentum.
The Z boson can be produced with a non-zero trasverse momentum, balanced by the amount of ISR that is emitted from the incoming partons, in fact, each split can give to the incoming parton a non-zero initial value of p_T and so the Z boson is created with a non-zero p_T before the introduction of the primordial k_T.

6.2 Primordial k_T tune

The parameters variation ranges used for the tune are the ones reported in the Table 6.1 while other parameters are set to CP5 values.

Parameter Name	Value
`BeamRemnants:primordialKThard`	$[0.5 - 5.0]$
`SpaceShower:pT0Ref`	$[0.5 - 5.0]$

Table 6.1: Variation ranges for the sampling used in the primordial k_T tune.

The number of samples is lower than the one used for the Underlying Event. The training set that has been used for the PerBin Model contains 160 MC runs and 200 for the Inverse Model, this is related to the lower number of parameters to tune. So, the operation of the tune was computationally faster: less Monte Carlo jobs to run.

Fig. 6.2 shows the distributions used to perform the tune. The distributions are from the [46] analysis at $\sqrt{s} = 13$ TeV:

- The Z boson production cross-section in Drell Yan measurements as a function of p_T^Z;
- The Z boson production cross-section in Drell Yan measurements as a function of ϕ_η^*.

Note that the region of interest is the low p_T region. The simulation of the whole spectrum requires the adoption of a merging scheme between the higher-order matrix element calculations and parton shower, as FxFx, which is computationally expensive. The strategy followed was the tuning of the low region, the one of interest, taking only the first five bins from each distribution.

The PerBin Model minimization results are displayed in Fig. 6.4. The blue line indicates the value of the χ^2/DoF as a function of the parameter value. The best estimation is marked by a solid red line while the dashed red lines indicate the errors, these results are reported in Table 6.3. Both the parameters are in a well-defined minimum so the minimization phase work properly.

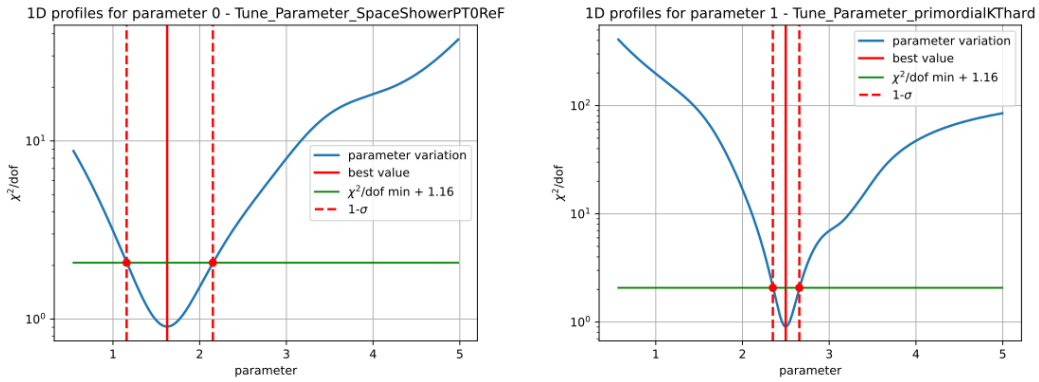


Figure 6.4: The output of the minimizer for the PerBin model in the primordial k_T tune. The left panel shows the result of the minimization for the `SpaceShower:pT0Ref` parameter while the right one `BeamRemnants:primordialKThard`. The blue line is the χ^2/DoF the best value is indicated by the solid red line while the errors are the ones indicated by the dashed red lines.

The Inverse model distribution of predictions for the best model is reported in Fig. 6.5: the best estimation for the parameter is indicated by the solid black line while the dashed black lines are the errors computed using the standard deviation. The hyperparameter space scanned to search for these best model is the same used above and reported in Table 5.2. At the end of the scanning, the best hyperparameters obtained are the ones reported in the Table 6.2.
The PerBin Model values and errors obtained as output from the tune are reported in Table 6.3. These are also compared to the ones obtained from the Inverse model. The value obtained from the two MCNNTUNES models are compatible with each other. The default values for CP5 are also reported and they are the ones inherited

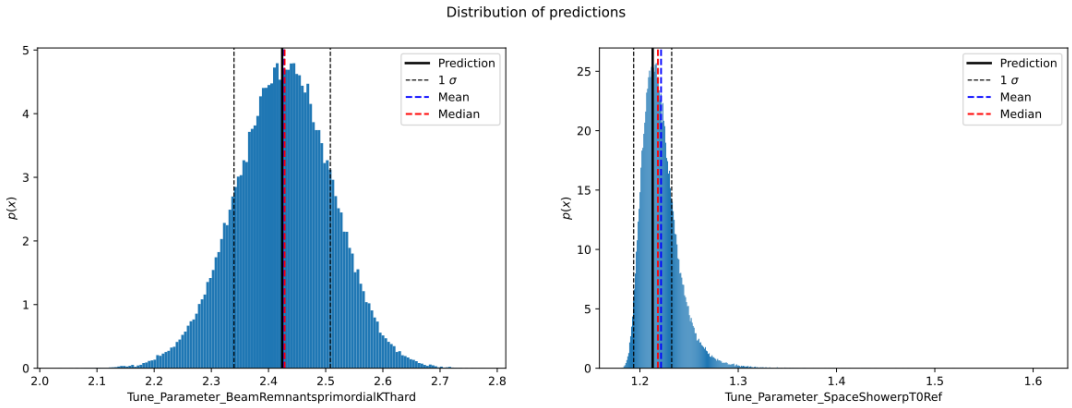


Figure 6.5: The prediction spread for the Inverse model in the primordial k_T tune. The left histogram refers to the `BeamRemnants:primordialKThard` parameter while the right one to the `SpaceShower:pT0Ref`. The predictions are indicated by the solid black vertical lines and the errors by the dashed black lines.

Hyperparameter	Value
Number of hidden layers	2
Units layers	[15, 17]
Activation function	sigmoid
Optimizer	rmsprop
Epochs	2000
Batch size	64

Table 6.2: Best hyperparameters model found for the primordial k_T tune with Inverse model.

by the Monash tune [47]. The values obtained by the tunes based on MCNNTUNES are quite different to the ones in CP5.

The overall results are displayed in Fig. 6.6. These new tunes describe better the low regions of the two spectra. These low regions are the ones actually tuned, the higher p_T region is only simulated using the value obtained from the tune of the first 5 bins in each distribution of Fig. 6.6. To simulate all the spectrum it has been employed the FxFx margin scheme in order to avoid double-counting and get a correct result from the simulation.

Parameter Name	PerBin	Inverse	Default CP5
`BeamRemnants:primordialKThard`	$2.5^{+0.2}_{-0.2}$	2.42 ± 0.08	1.8
`SpaceShower:pT0Ref`	$1.6^{+0.5}_{-0.5}$	1.21 ± 0.02	2.0

Table 6.3: Results obtained from the PerBin and the Inverse model in the tuning of the low region of Z boson production spectra. They are compared to the default for CP5 (inherited from Monash tune.)

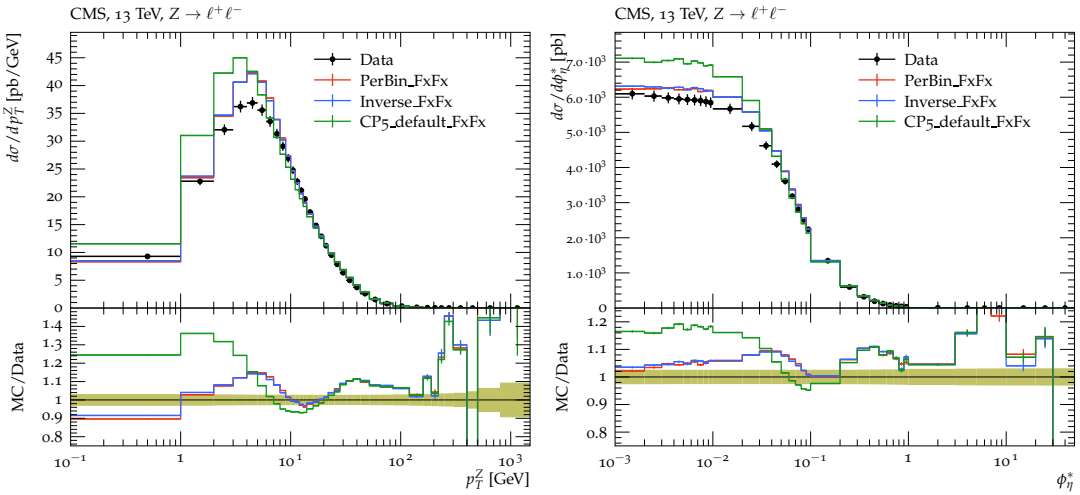


Figure 6.6: The results obtained from the tune of the primordial k_T. The left figure shows the Z boson production cross-section as a function of the p_T^Z. The red line refers to the PerBin model tune, the blue line to the Inverse Model and the green line is the CP5 default tune. The black points are the experimental data. The colored vertical lines the statistical uncertainties. It is clear that the two tunes based on MCNNTUNES better describe the low region with respect to the original CP5.

6.3 Primordial k_T tune vs MPI

As discussed above primordial k_T is not the only source of non zero transverse momentum in LO Z boson production. This is the reason why also the parameter that controls the amount of Initial State Radiation is tuned here. As described above, the ISR can leads to a non zero initial transverse momentum a larger amount of ISR (low `SpaceShower:pT0Ref`) is related to more splitting occurring in the evolution of the incoming partons before these partons undergo the hard scattering in which the Z boson is produced. But this is not the only process that can lead to a non-zero initial transverse momentum, there are also the MPI that can contribute to this initial transverse momentum. It is not an easy task to understand the effect of MPI on the p_T of the produced Z boson.

Fig. 6.7 shows the effect of the MPI on the distributions used for the tune. The blue line is obtained setting `PartonLevel:MPI=off` in PHYTIA8 configuration. It has been performed a calculation employing only a LO calculation, so the FxFx margin scheme is not required for this simulation. The high region of the distribution is not well described by the LO calculation but in this case the effect of the MPI are expected only in the low-p_T^Z region.

The main parameter that controls the number of parton interactions in a single hadron-hadron collision is the `MultipartonInteractions:pT0Ref` parameter. It is clear that the first 2 or 3 bins of the left distribution of Fig. 6.8 are sensitive to the variation of this parameter. A future thing to do is to investigate the effects of these MPI in more detail and perform a more general tune including them in this description.

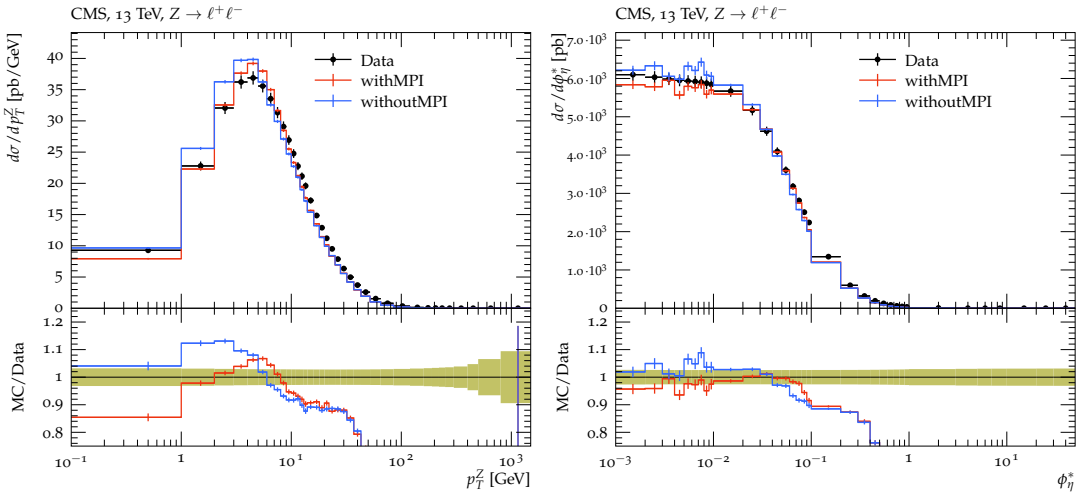


Figure 6.7: The image shows that MPI have an effect on the two distribution. The blue line is the result without the MPI (`PartonLevel:MPI=off`) and the red line with MPI. FxFx has not been used here: the high regions of the spectra are not well described by the simulation.

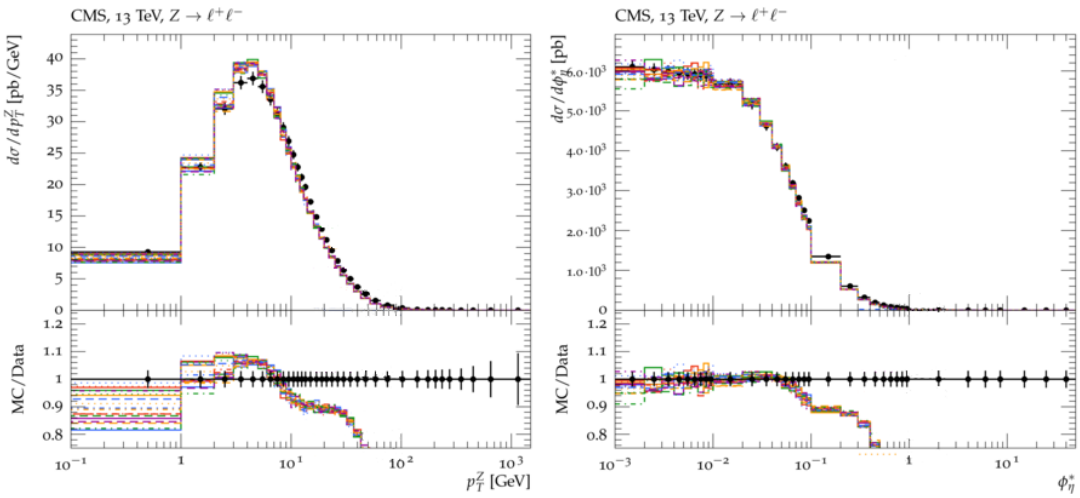


Figure 6.8: The image shows that `MPI:pT0Ref` affect the first bins of the left distribution. The black points are the experimental data while the various colored lines are the simulations with different values for the parameter that controls the `MPI:pT0Ref`. FxFx has not been used here: the high regions of the spectra are not well described by the simulation.

Conclusions

In this work, the use of the MCNNTUNES as a tuning tool for the HEP Monte Carlo generator PYTHIA8 has been tested. Firstly, in order to validate it, the results of MCNNTUNES were compared to those of an already existing CMS tune for the underlying event parameters: CP5. The work started with a test phase with the variation of only two parameters: `MultipartonInteractions:pT0Ref` and `MultipartonInteractions:ecmPow`. Both MCNNTUNES models perform very well in this test phase.
After this test, the number of parameters was extended to five as in the CP5 implementation. So, the parameters `MultipartonInteractions:coreRadius`, `MultipartonInteractions:coreFraction` and `ColorReconnection:range` have been added to the two parameters already tuned. In this phase we obtain very good results from the PerBin model. In particular the low-p_T regions of the charged particle density and charged particle p_T sum distributions are better described than in the original CP5 tune. In order to investigate the cause of a discrepancy in pseudorapidity distributions of charged hadron production and non-single diffractive event selections, a reweighting technique was introdcued. This leads to overall results more similar to the CP5 ones.
On the other hand the Inverse model results are not as solid, in this case the parameters found by MCNNTUNES cannot reproduce the observed experimental data. This can be related to the complexity of the operation performed by this model: to learn the function that describes the generator respond and try to invert it. The complexity of this procedure increases a lot with the number of parameters.
Once the tool has been validated, it is used to improve the description of the distributions of Z boson production events in the low transverse momentum region. To do that the effect of the parameters `BeamRemnants:primordialKThard` and `SpaceShower:pT0Ref` on these distributions was investigated. CMS tunes inherit the values of such parameters from the Monash 2013 tune and no attempt is made in CP5 to fit them. The new tune presented here for these two parameters was performed with both models offered by MCNNTUNES and the results obtained are compatible with each other.
In the future, the correlation between the MPI and the tuning of the primordial k_T has to be investigated. In the last chapter it was showed that the effect of MPI is not negligible in the low transverse momentum region.
Overall MCNNTUNES can be considered a valid tool for future tunes and it is shown to be an alternative to the already existing and consolidated tool PROFESSOR.

Appendix A

Test with two free parameters for the Underlying Event

Here are reported all the distributions obtained by the test on two parameters:

- `MultipartonInteractions:pT0Ref`
- `MultipartonInteractions:ecmPow`

We compare the results we get from PerBin Model and from Inverse Model with the CP5 ones.
The value we get are also reported in the Table A.1. The PerBin Model errors are not reported because there was not a proper errors estimation correctly implemented yet for this model.

Parameter	PerBin	Inverse	CP5 (down & up)
`MPI:pT0Ref`	1.46064	1.43 ± 0.14	$1.41 - 1.46$
`MPI:ecmPow`	0.04771	0.0298 ± 0.0095	0.03

Table A.1: Result PerBin model and Inverse Model in two parameter variation test. The value are compared to the upper and lower limit for CP5.

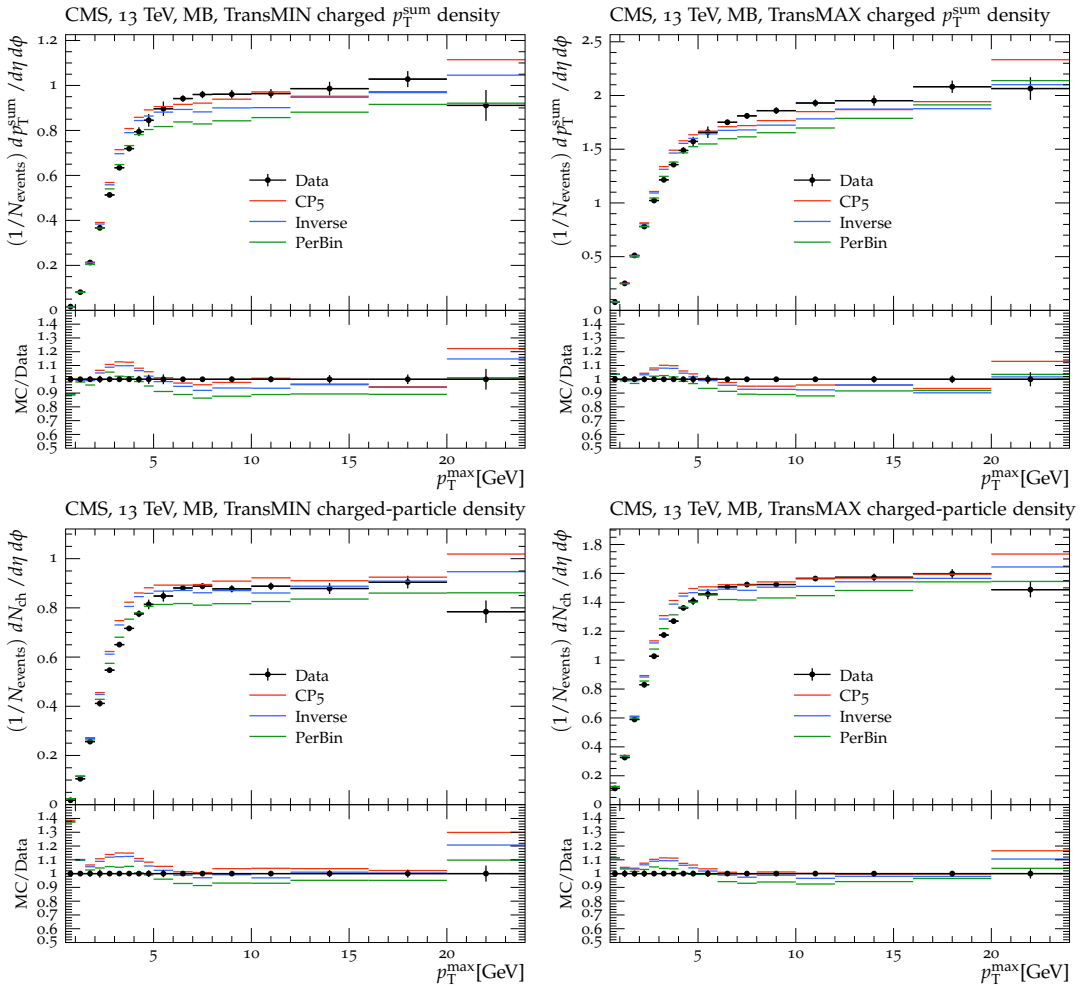


Figure A.1: In this figure the data from the $\sqrt{s} = 13$ TeV CMS analysis [33] that show the transMAX charged particle density (upper left) and the charged p_T-sum density (upper right); the transMIN charged particle density (lower left) and the charged p_T-sum density (lower right) as a function of the transverse momentum of the leading charged particle. The CP5 tune is compared to our test tune using the PerBin Model and Inverse Model.

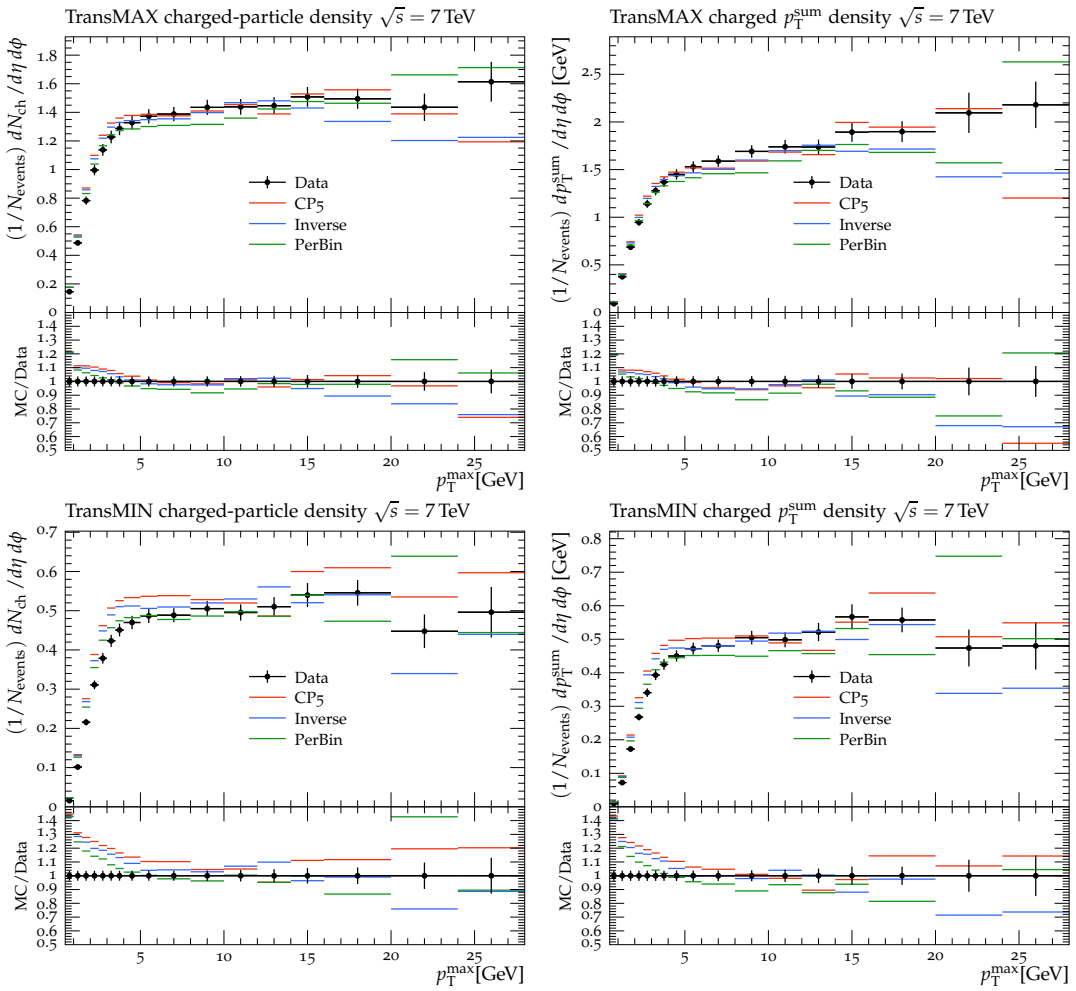


Figure A.2: Here the data at $\sqrt{s} = 7$ TeV from the CMS analysis [34] for the transMAX charged particle density (upper left) and the charged p_T-sum (upper left); the transMIN charged particle density (lower left) and the charged $p-T$-sum are displayed as a function of the transverse momentum of the leading object.

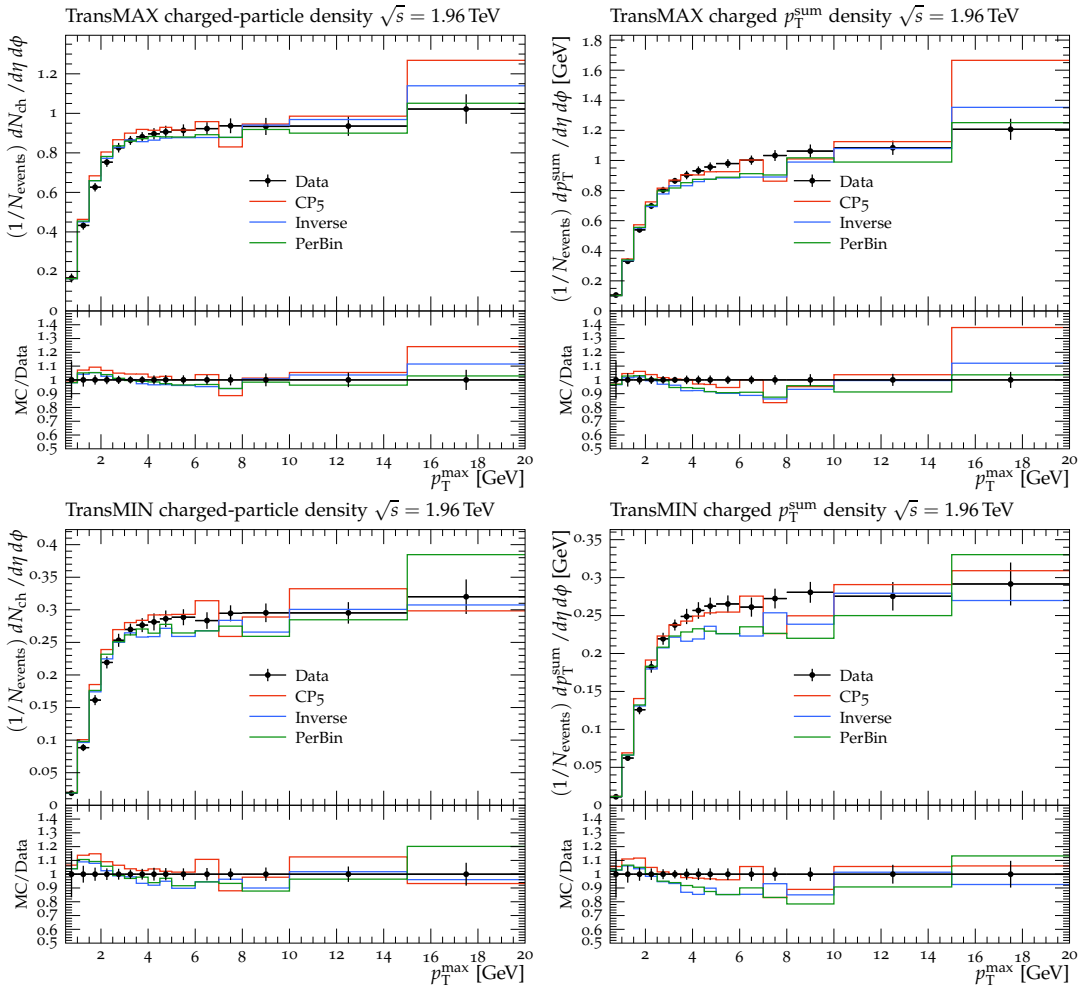


Figure A.3: The transMAX charged particle density (upper left) and the charged p_T-sum (upper left); the transMIN charged particle density (lower left) and the charged $p-T$-sum from the CDF analysis at $\sqrt{s} = 1.96$ TeV in proton-antiproton collisions [35].

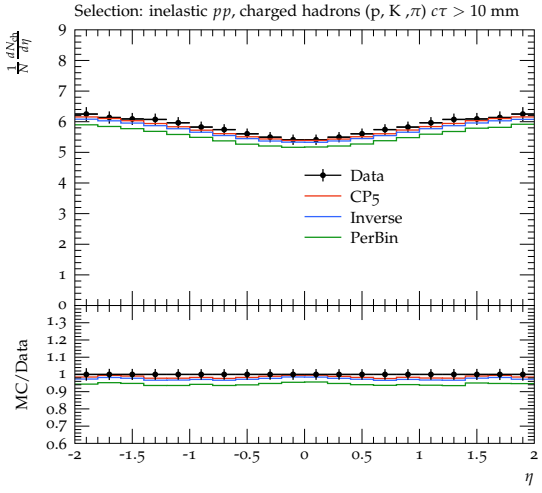


Figure A.4: In this figure is shown the last distribution we use for the tune from the CMS analysis at $\sqrt{s} = 13$ TeV [40]. The pseudorapidity distribution ($|\eta| < 2$) for the charged hadron density in an inelastic proton-proton scattering selection.

Bibliography

[1] Steven Weinberg. A model of leptons, Nov 1967. URL https://link.aps.org/doi/10.1103/PhysRevLett.19.1264.

[2] P.A. Zyla et al. Review of Particle Physics. *PTEP*, 2020(8):083C01, 2020. doi: 10.1093/ptep/ptaa104. and 2021 update.

[3] H. Abramowicz et al. Combination of measurements of inclusive deep inelastic $e^{\pm}p$ scattering cross sections and QCD analysis of HERA data. *Eur. Phys. J. C*, 75(12):580, 2015. doi: 10.1140/epjc/s10052-015-3710-4.

[4] R. P. Feynman. The behavior of hadron collisions at extreme energies. *Conf. Proc. C*, 690905:237–258, 1969.

[5] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An introduction to pythia 8.2, Jun 2015. ISSN 0010-4655. URL http://dx.doi.org/10.1016/j.cpc.2015.01.024.

[6] Albert M Sirunyan et al. Extraction and validation of a new set of CMS PYTHIA8 tunes from underlying-event measurements. *Eur. Phys. J. C*, 80(1):4, 2020. doi: 10.1140/epjc/s10052-019-7499-4.

[7] Marco Lazzarin, Simone Alioli, and Stefano Carrazza. MCNNTUNES: Tuning Shower Monte Carlo generators with machine learning. *Comput. Phys. Commun.*, 263:107908, 2021. doi: 10.1016/j.cpc.2021.107908.

[8] Andy Buckley, Hendrik Hoeth, Heiko Lacker, Holger Schulz, and Jan Eike von Seggern. Systematic event generator tuning for the LHC. *Eur. Phys. J. C*, 65: 331–357, 2010. doi: 10.1140/epjc/s10052-009-1196-7.

[9] J. D. Bjorken. Asymptotic Sum Rules at Infinite Momentum. *Phys. Rev.*, 179: 1547–1553, 1969. doi: 10.1103/PhysRev.179.1547.

[10] Sidney D Drell and Tung-Mow Yan. Partons and their applications at high energies, 1971. ISSN 0003-4916. URL https://www.sciencedirect.com/science/article/pii/0003491671900716.

[11] L N Lipatov. The parton model and perturbation theory, 1975. URL http://cds.cern.ch/record/400357.

[12] Vladimir Naumovich Gribov and L N Lipatov. Deep inelastic ep scattering in perturbation theory, 1972. URL https://cds.cern.ch/record/427157.

[13] G. Altarelli and G. Parisi. Asymptotic freedom in parton language, 1977. ISSN 0550-3213. URL https://www.sciencedirect.com/science/article/pii/0550321377903844.

[14] Yuri L. Dokshitzer. Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics., 1977.

[15] Richard D. Ball, Valerio Bertone, Stefano Carrazza, Luigi Del Debbio, Stefano Forte, Patrick Groth-Merrild, Alberto Guffanti, Nathan P. Hartland, Zahari Kassabov, José I. Latorre, Emanuele R. Nocera, Juan Rojo, Luca Rottoli, Emma Slade, and Maria Ubiali. Parton distributions from high-precision collider data. *The European Physical Journal C*, 77(10), Oct 2017. ISSN 1434-6052. doi: 10.1140/epjc/s10052-017-5199-5. URL http://dx.doi.org/10.1140/epjc/s10052-017-5199-5.

[16] F. Bloch and A. Nordsieck. Note on the radiation field of the electron, Jul 1937. URL https://link.aps.org/doi/10.1103/PhysRev.52.54.

[17] Toichiro Kinoshita. Mass singularities of feynman amplitudes, 1962. URL https://doi.org/10.1063/1.1724268.

[18] T. D. Lee and M. Nauenberg. Degenerate systems and mass singularities, Mar 1964. URL https://link.aps.org/doi/10.1103/PhysRev.133.B1549.

[19] J M Campbell, J W Huston, and W J Stirling. Hard interactions of quarks and gluons: a primer for lhc physics, Dec 2006. ISSN 1361-6633. URL http://dx.doi.org/10.1088/0034-4885/70/1/R02.

[20] Stefano Frixione and Bryan R Webber. Matching nlo qcd computations and parton shower simulations. *Journal of High Energy Physics*, 2002(06):029–029, Jun 2002. ISSN 1029-8479. doi: 10.1088/1126-6708/2002/06/029. URL http://dx.doi.org/10.1088/1126-6708/2002/06/029.

[21] Stefano Frixione, Paolo Nason, and Bryan R Webber. Matching nlo qcd and parton showers in heavy flavour production. *Journal of High Energy Physics*, 2003(08):007–007, Aug 2003. ISSN 1029-8479. doi: 10.1088/1126-6708/2003/08/007. URL http://dx.doi.org/10.1088/1126-6708/2003/08/007.

[22] Stefano Frixione and Bryan R. Webber. The mc@nlo 3.1 event generator, 2005.

[23] Rikkert Frederix and Stefano Frixione. Merging meets matching in MC@NLO. *JHEP*, 12:061, 2012. doi: 10.1007/JHEP12(2012)061.

[24] Sparsh Navin. Diffraction in Pythia. 5 2010.

[25] FRANK SIEGERT. Monte-carlo event generation for the lhc, 2010. URL http://etheses.dur.ac.uk/484/.

[26] CMS Colaboration. CMS PYTHIA 8 colour reconnection tunes based on underlying-event data. 5 2022.

[27] B. Andersson, G. Gustafson, G. Ingelman, and T. Sjöstrand. Parton fragmentation and string dynamics. *Physics Reports*, 97(2):31–145, 1983. ISSN 0370-1573. doi: https://doi.org/10.1016/0370-1573(83)90080-7. URL https://www.sciencedirect.com/science/article/pii/0370157383900807.

[28] Torbjorn Sjostrand. Jet Fragmentation of Nearby Partons. *Nucl. Phys. B*, 248: 469–502, 1984. doi: 10.1016/0550-3213(84)90607-2.

[29] Christian Bierlich et al. Robust Independent Validation of Experiment and Theory: Rivet version 3. *SciPost Phys.*, 8:026, 2020. doi: 10.21468/SciPostPhys.8.2.026.

[30] Matt Dobbs and Jørgen Beck Hansen. The hepmc c++ monte carlo event record for high energy physics. *Computer Physics Communications*, 134(1): 41–46, 2001. ISSN 0010-4655. doi: https://doi.org/10.1016/S0010-4655(00)00189-2. URL https://www.sciencedirect.com/science/article/pii/S0010465500001892.

[31] YODA. Yoda. URL https://yoda.hepforge.org/.

[32] Tai Sakuma and Thomas McCauley. Detector and Event Visualization with SketchUp at the CMS Experiment. *J. Phys. Conf. Ser.*, 513:022032, 2014. doi: 10.1088/1742-6596/513/2/022032.

[33] Underlying Event Measurements with Leading Particles and Jets in pp collisions at $\sqrt{s} = 13$ TeV. Technical report, CERN, Geneva, 2015. URL https://cds.cern.ch/record/2104473.

[34] Measurement of the Underlying Event Activity at the LHC at 7 TeV and Comparison with 0.9 TeV. Technical report, CERN, Geneva, 2012. URL http://cds.cern.ch/record/1478982.

[35] Timo Antero Aaltonen et al. Study of the energy dependence of the underlying event in proton-antiproton collisions. *Phys. Rev. D*, 92(9):092009, 2015. doi: 10.1103/PhysRevD.92.092009.

[36] Vardan Khachatryan et al. CMS Tracking Performance Results from Early LHC Operation. *Eur. Phys. J. C*, 70:1165–1192, 2010. doi: 10.1140/epjc/s10052-010-1491-3.

[37] Gavin P. Salam and Gregory Soyez. A Practical Seedless Infrared-Safe Cone jet algorithm. *JHEP*, 05:086, 2007. doi: 10.1088/1126-6708/2007/05/086.

[38] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The anti-k_t jet clustering algorithm. *JHEP*, 04:063, 2008. doi: 10.1088/1126-6708/2008/04/063.

[39] Albert M. Sirunyan et al. Measurement of charged particle spectra in minimum-bias events from proton–proton collisions at $\sqrt{s} = 13\,\text{TeV}$. *Eur. Phys. J. C*, 78 (9):697, 2018. doi: 10.1140/epjc/s10052-018-6144-y.

[40] Vardan Khachatryan et al. Pseudorapidity distribution of charged hadrons in proton-proton collisions at $\sqrt{s} = 13$ TeV. *Phys. Lett. B*, 751:143–163, 2015. doi: 10.1016/j.physletb.2015.10.004.

[41] Serguei Chatrchyan et al. Measurement of the underlying event in the Drell-Yan process in proton-proton collisions at $\sqrt{s} = 7$ TeV. *Eur. Phys. J. C*, 72: 2080, 2012. doi: 10.1140/epjc/s10052-012-2080-4.

[42] A. M. Sirunyan et al. Measurement of the underlying event activity in inclusive Z boson production in proton-proton collisions at $\sqrt{s} = 13$ TeV. *JHEP*, 07:032, 2018. doi: 10.1007/JHEP07(2018)032.

[43] Albert M. Sirunyan et al. Study of the underlying event in top quark pair production in pp collisions at 13 TeV. *Eur. Phys. J. C*, 79(2):123, 2019. doi: 10.1140/epjc/s10052-019-6620-z.

[44] Florian Bechtel. *The underlying event in proton-proton collisions*. PhD thesis, Hamburg U., 2009.

[45] A. Banfi, S. Redford, M. Vesterinen, P. Waller, and T. R. Wyatt. Optimisation of variables for studying dilepton transverse momentum distributions at hadron colliders. *Eur. Phys. J. C*, 71:1600, 2011. doi: 10.1140/epjc/s10052-011-1600-y.

[46] Albert M Sirunyan et al. Measurements of differential Z boson production cross sections in proton-proton collisions at $\sqrt{s} = 13$ TeV. *JHEP*, 12:061, 2019. doi: 10.1007/JHEP12(2019)061.

[47] Peter Skands, Stefano Carrazza, and Juan Rojo. Tuning PYTHIA 8.1: the Monash 2013 Tune. *Eur. Phys. J. C*, 74(8):3024, 2014. doi: 10.1140/epjc/s10052-014-3024-y.

[48] Stefano Carrazza and Marco Lazzarin. N3pdf/mcnntunes: mcnntunes 0.1.0, October 2020. URL `https://doi.org/10.5281/zenodo.4071125`.

[49] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[50] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. 1958. URL `https://doi.org/10.1037/h0042519`.

[51] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(91)90009-T. URL `https://www.sciencedirect.com/science/article/pii/089360809190009T`.

[52] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(05)80131-5. URL `https://www.sciencedirect.com/science/article/pii/S0893608005801315`.

[53] Nikolaus Hansen. The CMA evolution strategy: A tutorial. *CoRR*, abs/1604.00772, 2016. URL `http://arxiv.org/abs/1604.00772`.

[54] G. Cowan. *Statistical data analysis*. Oxford University Press, USA, 1998.

[55] Richard D. Ball et al. Parton distributions from high-precision collider data. *Eur. Phys. J. C*, 77(10):663, 2017. doi: 10.1140/epjc/s10052-017-5199-5.

[56] Albert M Sirunyan et al. Measurement of differential cross sections for Z boson production in association with jets in proton-proton collisions at $\sqrt{s} = 13$ TeV. *Eur. Phys. J. C*, 78(11):965, 2018. doi: 10.1140/epjc/s10052-018-6373-0.