

---

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 Dessenà**

Correlatore:

**Prof. Marco Monteno**

Controrelatore:

**Prof. Paolo Torrielli**

Anno Accademico 2021/2022



## 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 the real data. Due to the high computational cost of running a generator, the tuning requires a parametrization-based approach: the state-of-art 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.

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 and than use it for a further tune. To perform this validation we try to reproduce an already existing tune for the description of the Underlying Event in hadron-hadron scatters using data collected in Minimum Bias observation. So, in the first part the parameters related to the Multi Parton Interactions and Color Reconnection, in PYTHIA event generator, have been tuned. The tune was performed employing data from the CDF and CMS experiments at different center-of-mass energy. The validation of the MCNNTUNES tool is performed in two steps: a first test with a restrict number of free parameters and the actual tune.

Once the tool has been validated, we employed it for 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.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Hadron-Hadron Scattering</b>	<b>5</b>
2.1	QCD factorization theorem . . . . .	5
2.2	Partonic cross-section . . . . .	7
2.2.1	Higher order calculations . . . . .	8
2.3	Parton Showers . . . . .	11
2.3.1	Merging parton showers and matrix element calculations . . . . .	13
2.4	Parton distribution functions . . . . .	14
2.5	A real proton-proton collision . . . . .	15
<b>3</b>	<b>Multiple Parton Interactions</b>	<b>17</b>
3.1	Basic Concepts . . . . .	17
3.2	Pythia8 Monte Carlo events generator . . . . .	21
3.2.1	Parton Shower . . . . .	21
3.2.2	Multiple Parton Interactions in Pythia . . . . .	21
3.2.3	Momentum and flavour conservation . . . . .	22
3.2.4	Impact Parameter Dependence . . . . .	22
3.2.5	Parton rescattering . . . . .	23
3.2.6	Interleaving of Multiple Interaction and Parton Shower . . . . .	24
3.2.7	Beam Beam Remnants and primordial $k_T$ . . . . .	25
3.2.8	Color Reconnection and Hadronization . . . . .	26
3.3	PYTHIA summary . . . . .	28
<b>4</b>	<b>CMS Detector, Observable to Study the Underlying Event and Existing Tunes</b>	<b>29</b>
4.1	Compact Muon Solenoid Detector . . . . .	29
4.2	Minimum Bias Measurements and Underlying Event topology . . . . .	31
4.2.1	Observables in Z production processes . . . . .	34
4.3	Previous Tune for the Underlying Event . . . . .	37
4.3.1	The distributions used for the Tune . . . . .	37
4.3.2	Pythia configuration and the tunes . . . . .	38
<b>5</b>	<b>Tune procedure and MCNNTUNES</b>	<b>39</b>
5.1	Parametrization-based approach . . . . .	40
5.2	Machine Learning and Neural Networks . . . . .	40
5.2.1	Neural Networks - Perceptron . . . . .	41
5.2.2	Feed-Forward Neural Networks . . . . .	41

5.3	MCNNNTUNES . . . . .	44
5.3.1	Sampling Phase and Training Set Generation . . . . .	45
5.3.2	Per Bin Model . . . . .	45
5.3.3	Negative aspects . . . . .	48
5.3.4	Inverse Model . . . . .	48
5.3.5	Weighrules in MCNNNTUNES . . . . .	50
<b>6</b>	<b>Our Tune for the Underlying Event</b>	<b>53</b>
6.1	Introduction . . . . .	53
6.2	First test: only two parameters variation . . . . .	53
6.2.1	Per Bin Model results . . . . .	54
6.2.2	Inverse Model results . . . . .	57
6.2.3	Overall results . . . . .	59
6.3	Our tune for the UE in Minimum Bias observations . . . . .	60
6.3.1	Per Bin Model results . . . . .	60
6.3.2	Inverse Model results . . . . .	64
6.3.3	Overall results . . . . .	65
<b>7</b>	<b>Tune for the Primordial <math>k_T</math></b>	<b>71</b>
7.1	Primordial $k_T$ and ISR effect on $p_T^Z$ . . . . .	71
7.2	Primordial $k_T$ tune . . . . .	73
7.3	Primordial $k_T$ tune vs MPI . . . . .	76
<b>Conclusions</b>		<b>79</b>
<b>A Test with two free parameters for the Underlying Events</b>		<b>81</b>
<b>Bibliography</b>		<b>85</b>

# Chapter 1

## Introduction

The *standard model of particle* (SM) is the theory that describe all the elementary particles and the fundamental interactions: electromagnetic, weak and the strong interaction. The standard model 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 the electromagnetic interaction 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 related to the strong interactions of quarks.

The particle contained in the SM are grouped in *fermions*, particles with semi-integer spin, and *bosons*, particles with integer spin. The fermions contained in the standard model are sub-divided into leptons and quarks. While the boson are the gauge particles (spin 1), i.e. the particles that mediate the interaction: photons, gluons,  $W$  and  $Z$  bosons. Lastly, the famous Higgs boson that is the scalar field of the theory (particle with spin 0). The masses of all the particles in the SM are derived by the interaction with this last particle by mean of the Higgs mechanism. Table 1.1 shows a summary of all the fundamental particles of the SM with discovery dates, masses and some other properties.

The discover of the Higgs boson in 2012 at the Large Hadron Collider (CERN) was the last ingredient that was missing in the SM. The Higgs boson was predicted by the theory but not yet discovered. This discovery confirm the theory and complete it.

In a high energy hadron-hadron collision all the interactions named above take place and participate in the creation of the complex final state observed in the experiments that investigate the fundamental particle physics. Obviously the dominating one is the strong interaction it is present in lots of processes

Historically the hard interaction was associated to the hadrons, but with the deep inelastic scattering experiments [3] it was proven that the hadrons are composite objects and than they cannot be fundamental particles, 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.

So the partons model leads to the introduction of the so called Parton Distribution Function (PDF) in order to describe the probability of finding a parton carrying a

Name	Discovered at	Mass	Charge	Colour	Spin	Lifetime
electron $e$	Cavendish Laboratory (1897)	0.511 MeV	-1	no	1/2	stable
muon $\mu$	Caltech & Harvard (1937)	105.66 MeV	-1	no	1/2	$2.2 \times 10^{-6}$ sec
tau $\tau$	SLAC (1976)	1776.82 MeV	-1	no	1/2	$2.9 \times 10^{-13}$ sec
electron neutrino $\nu_e$	Savannah River Plant (1956)	< 2 eV	0	no	1/2	stable
muon neutrino $\nu_\mu$	Brookhaven (1962)	< 0.19 MeV	0	no	1/2	stable
tau neutrino $\nu_\tau$	Fermilab (2000)	< 18.2 MeV	0	no	1/2	stable
up quark $u$	SLAC (1968)	2.3 MeV	2/3	yes	1/2	stable
down quark $d$	SLAC (1968)	4.8 MeV	-1/3	yes	1/2	stable
charm quark $c$	Brookhaven & SLAC (1974)	1.275 GeV	2/3	yes	1/2	$1.1 \times 10^{-12}$ sec
strange quark $s$	Manchester University (1947)	95 MeV	-1/3	yes	1/2	$1.24 \times 10^{-8}$ sec
top quark $t$	Fermilab (1995)	173.21 GeV	2/3	yes	1/2	$4.2 \times 10^{-25}$ sec
bottom quark $b$	Fermilab (1977)	4.18 GeV	-1/3	yes	1/2	$1.3 \times 10^{-12}$ sec
photon $\gamma$	Washington University (1923)	0	0	no	1	stable
gluon $g$	DESY (1979)	0	0	yes	1	stable
$W$ boson	CERN (1983)	80.385 GeV	$\pm 1$	no	1	$3 \times 10^{-25}$ sec
$Z$ boson	CERN (1983)	91.1876 GeV	0	no	1	$3 \times 10^{-25}$ sec
Higgs boson $H$	CERN (2012)	125.7 GeV	0	no	0	$1.56 \times 10^{-22}$ sec

**Figure 1.1:** All the fundamental particles of the standard model. Figure from [2]

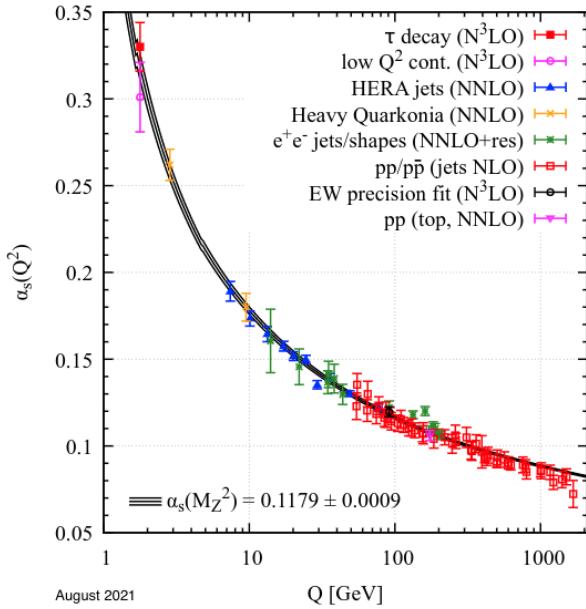
fraction of the proton momentum.

A peculiarity of the strong interaction theory QCD respect to the QED is that the QCD group  $SU(3)$  is not an abelian group this reflect in the color charge of the vector boson of the theory. In fact, the gluons are color charged and can interact each other. While in the QED this was not possible in fact the photons are not cahrged particles. This peculiarity leads to a different behavior of the QCD coupling constant  $\alpha_s$ .

This peculiarity leads to the evolution of the coupling constant, shown in Fig. 1.2, on the momentum scale  $Q^2$  of the interaction observed in nature. When  $Q^2 \gg 1$  GeV (short distance) the  $\alpha_s(Q^2)$  is small and so the quark and the gluon are weakly coupled, this is called *asymptotic freedom*. In this limit the processes can be evaluate using perturbation theory. This type of processes are refered as *hard QCD* processes, where the hard attribute is related to the high energy scale of the interaction.

On the other hand, when  $Q^2$  is small (large distance), the value of the coupling constant is large, this leads to color confinement. Color confinement is the reason why we cannot observe free quarks in the experiments but as the energy scale of the interaction decrease the quarks are confined into hadrons. These type of processes where the energy scale is low are refered as *soft QCD* processes. Soft QCD processes cannot be described by perturbation theory and are not well understood starting from fundamental principles. The soft QCD processes are clearly an important fraction of the processes we have to describe in order to understand what is going on in real proton-proton collision. A clear example of the importance of the soft processes is the hadronization: once the partons inside the protons have interact the products of the interaction go toward the detector after a tiny distance all the quarks are bounded in hadrons. The hadronization is present in every collision, it is clear that we have to take it into account. So, the use of Monte Carlo simulations to describe these processes is required to explain the experimental data we get.

In this thesis we are going to focus on this aspect the Monte Carlo simulation have to describe the real data very well if we want to use it in experiments. The procedure



**Figure 1.2:** evolution of the coupling constant with the energy scale of the interaction [5]

to adapt the generator parameters values in order to reproduce the data correctly goes under the name of *tune*. The first chapter of this thesis is going to introduce the theoretical aspect in the description of an hadron-hadron collision. We start from the description of hard event using perturbation theory and than we move on the description of the all-order approaches (e.g. parton showers implemented in Monte Carlo generator). On Chapter 3 we focus on the main processes described by the Monte Carlo generators and in particular in PYTHIA8. Some particular attention is give to the Multi Parton Interaction, the Color Reconnection, the Primordial  $k_T$ , the Initial and Final State Radiations description. These are the main contribution we are focusing on in the tuning of the underlying event and minimum bias observables and of the  $Z$  boson production cross section at low transverse momentum of the  $Z$  boson. All the observables we are going to use in the tunes are described in Chapter 4 with an introduction on the existing CMS tunes called CP and a number from 1 to 5 that are the starting point for our work. The Chapter 5 describe the basic concepts in the tune procedure in particular on the parametrization-based approach used for the tuning, in this chapter we also discuss the main tool we are going to use that is called MCNTUNES it implement a tuning method based on machine learning. Our tunes for the underlying event and for the  $Z$  boson spectrum are described respectively in Chapter 6 and Chapter 7.



# Chapter 2

## Hadron-Hadron Scattering

In a high energy proton-proton collision we can have either soft or hard processes. Most of the time the hard processes are accompanied by soft interactions, occurring along the hadron interaction. While the hard processes 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, 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 approaches.

This chapter gives a theoretical introduction on some basic elements of perturbative QCD theory: the QCD factorization theorem, the fixed-order approach in perturbative QCD calculations, and finally the all-orders approaches (e.g. the parton showers).

### 2.1 QCD factorization theorem

The fundamental idea for the description of a hadron-hadron collision is that hadrons are made-up from 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, given a sufficiently high energy scale for the interaction.

This idea was developed in the framework of the deep inelastic scattering, and the Bjorken scaling observation confirms it [6]. The generalization of this concept leads to the QCD factorization theorem.

The factorization theorem was introduced first by Drell and Yan [7]. 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:

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

Where  $X$  is a partonic/leptonic state and  $a(b)$  a quark or an antiquark in the hadron  $A(B)$ . The  $\sigma_{AB}$  is the hadronic cross-section while  $\hat{\sigma}_{ab \rightarrow X}$  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  $A$  ( $B$ ) that is dependent on  $x_a$  ( $x_b$ ) that is the momentum fraction of the total momentum of the hadron  $A$  ( $B$ ) carried

by the parton. This is valid in the "scaling" limit:

$$s \rightarrow \infty, \quad \frac{M_X}{\sqrt{s}} = \text{finite} ; \quad (2.2)$$

where the center-of-mass energy grows to infinity but with a fixed ratio of the X state invariant mass and the center-of-mass energy.

A problem arises from the perturbative corrections from real and virtual gluon emission, in particular from the collinear gluon emissions. These contributions take to a logarithmic divergence (spoiling the convergence of the perturbative expansion). These dependencies can be absorbed by the parton distribution functions (that evolve according the DGLAP equations). This results in the scaling violation of the parton distribution functions

$$f_{a/A}(x_a) \rightarrow f_{a/A}(x_a, Q^2) , \quad (2.3)$$

i.e. the parton distribution function depend not only on the momentum fraction  $x$ , but also on the energy scale  $Q^2$  of the interaction.

So, we can rewrite the factorization theorem in Eq. 2.1 as:

$$\sigma_{AB} = \int dx_a dx_b f_{a/A}(x_a, Q^2) f_{b/B}(x_b, Q^2) \hat{\sigma}_{ab \rightarrow X} . \quad (2.4)$$

At this point the *finite* corrections to the leading-logarithmic cross section in the perturbative expansion are specific for each process (i.e. they are not universal, but they are process dependent). This leads in the Eq. 2.4 to the following series in  $\alpha_s$  (the coupling constant of QCD theory):

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

In Eq. 2.5 two scales enter the formula:

- The *factorization scale*  $\mu_F$ : this scale is related to the resolution with which the hadron is being probed, separates long- and short- distance physics processes.
- The *renormalization scale*  $\mu_R$ : the scale at which is evaluated the strong coupling constant  $\alpha_s$ . The dependence of  $\alpha_s$  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 higher-order corrections to the cross section in Eq. 2.5 are important because they lead to an higher accuracy in the cross section prediction, gradually reducing the dependencies on  $\mu_R$  and  $\mu_F$ . In absence of an all-order prediction, a choice for the two scales have to be taken. Typically, the scales are assumed to be equal: in the Drell-Yan process the standard choice is  $\mu_F = \mu_R = M$ , with  $M$  the lepton-pair mass [8]; other cases are the invariant masses of  $Z$ -boson, top quark or jet transverse energy to study [8] the production cross sections for  $Z$ -bosons, top quarks and large  $E_T$  jets respectively.

The parton distribution functions used to describe a hard scattering are solution to the DGLAP (Dokshitzer–Gribov–Lipatov–Altarelli–Parisi) equations [9, 10, 11,

[12]

$$\mu_F^2 \frac{\partial f_{i/H}(x, \mu_F^2)}{\partial \mu_F^2} = \sum_j \frac{\alpha_s(\mu_F^2)}{2\pi} \int_x^1 \frac{dz}{z} P_{i \rightarrow j}(z) f_{j/p}\left(\frac{x}{z}, \mu_F^2\right) , \quad (2.6)$$

where  $P_{i \rightarrow j}$  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  $\alpha_s$ :

$$P_{i \rightarrow j}(x, \alpha_s) = P_{i \rightarrow j}^{(0)}(x) + \frac{\alpha_s}{2\pi} P_{i \rightarrow j}^{(1)}(x) + \dots . \quad (2.7)$$

This procedure has been used to calculate cross-sections for many Standard Model processes in  $p\bar{p}$  and  $pp$  scattering respectively at Tevatron and LHC energies as shown in Fig. 2.1.

The parton distribution functions dependence on  $Q^2$  can be derived theoretically via the DGLAP equations. Instead, the  $x$  dependence is obtained fitting the deep-inelastic and other hard-scattering processes experimental data. The experimental coverage in  $(x, Q^2)$ -plane is shown in Fig. 2.2 where is also underlined the relationship between  $(x, Q^2)$  and kinematic variables in Drell-Yan processes for a final state with invariant mass  $M$  and rapidity  $y$  (further details in section 2 of [8]). Assuming that the factorization scale  $Q$  is equal to  $M$  (The reference center of mass energy is 13 TeV), for two incoming partons with four-momentum respectively  $p_1$  and  $p_2$ , the relations of their  $x$  (labeled as  $x_1$  and  $x_2$ ) with  $s$ ,  $y$  and  $M$  are:

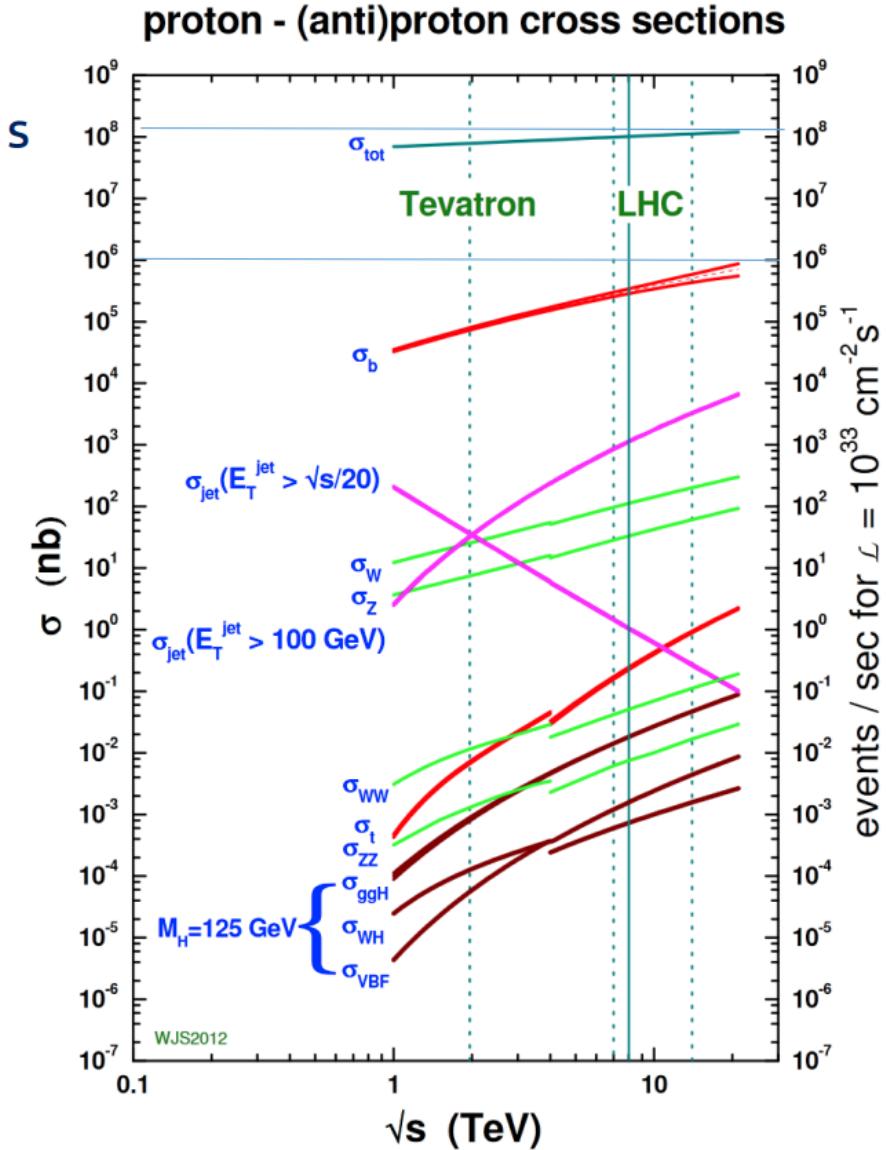
$$\begin{aligned} p_1^\mu &= \frac{\sqrt{s}}{2}(x_1, 0, 0, x_1) \\ p_2^\mu &= \frac{\sqrt{s}}{2}(x_2, 0, 0, -x_2) \end{aligned} \implies x_1 = \frac{M}{\sqrt{s}} e^y \quad x_2 = \frac{M}{\sqrt{s}} e^{-y} , \quad (2.8)$$

where  $s = (p_1^\mu + p_2^\mu)^2$ . For example the figure shows that the production of a final state with invariant mass  $M = 100$  GeV and rapidity  $y = 2$  is given by the interaction of two hadrons with  $x_1 \approx 0.05$  and  $x_2 \approx 0.001$  with  $Q^2 = 10^4$  GeV $^2$

## 2.2 Partonic cross-section

Partonic cross section as seen in Eq. 2.1 is one of the fundamental ingredients in our recipe for the description of hadron-hadron interactions. It can be calculated as a perturbative expansion in  $\alpha_s$  from QCD first principles using quantum field theory. 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, we can already encounter some divergence that can be avoided imposing some restrictions on the phase space.

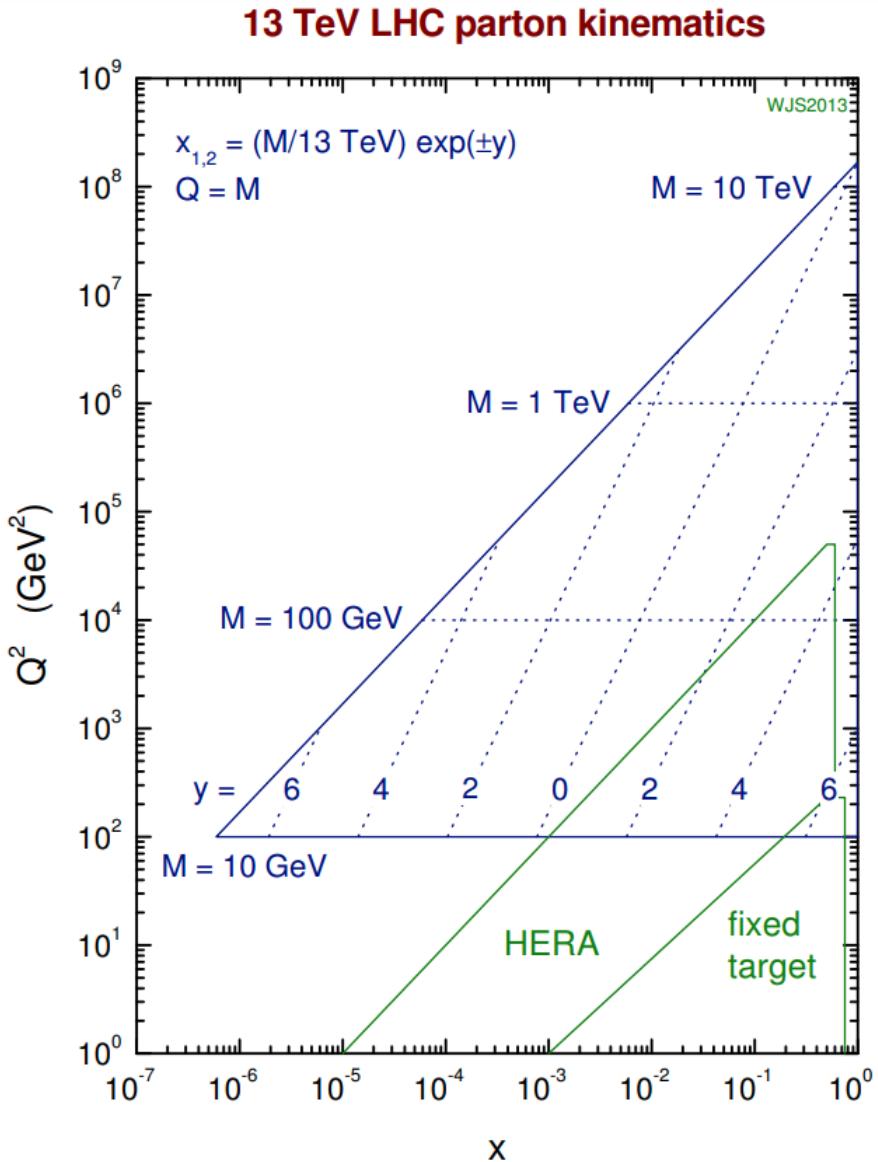


**Figure 2.1:** Next-to-leading order cross sections at Tevatron and LHC colliders energies (the splitting are at the transition between  $p\bar{p}$  and  $pp$  cross section). Figure from [13]

### 2.2.1 Higher order calculations

The LO calculation can describe broad feature 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 process 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  $\alpha_s$ . This contribution can arise in two different ways:

- **Virtual:** internal lines in the Feynmann diagram, see Fig. 2.3a, that don't originate a real particle in the initial or in the final state (loops);



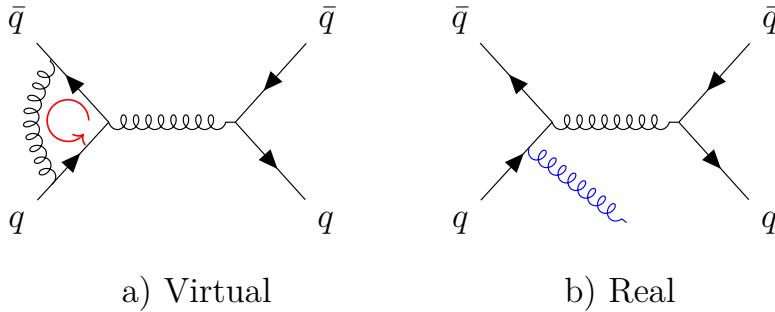
**Figure 2.2:** Graphical representation of the parton  $(x, Q^2)$  variables coverage of different experiments. For LHC these  $x$  and  $Q^2$  are related to the kinematic variables  $y$  and  $M$ . Figure from [13]

- **Real:** external lines int the Feynmann diagram, see Fig. 2.3b, here the particle is real and participate to the initial or final state observed (real particles).

Virtual corrections contains infrared divergences, arising by integrating on the loop circulating momentum, that cancel against infrared singularities given by collinear or soft emissions [14, 15, 16].

A common strategy for the renormalization is dimensional regularization: it consists into 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  $\epsilon$ . Then, the limit  $\epsilon \rightarrow 0$  is taken after the divergences have cancelled.

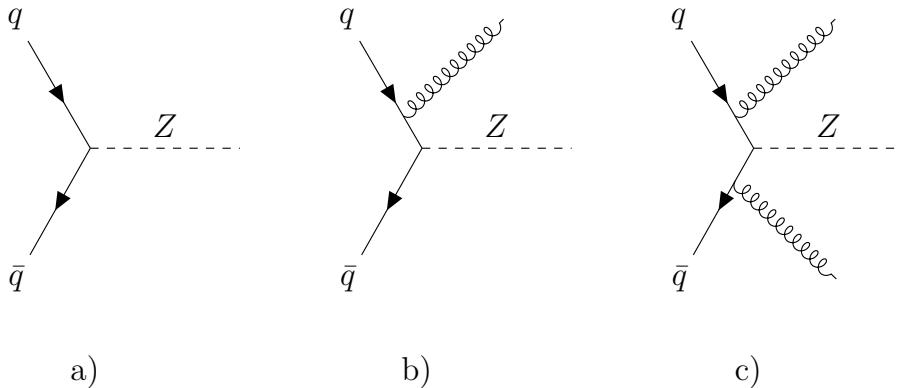
This NLO calculation with regularization allows to extend the treatment of these integrals up to zero transverse momentum.



**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, participate to the initial or final state of the process.

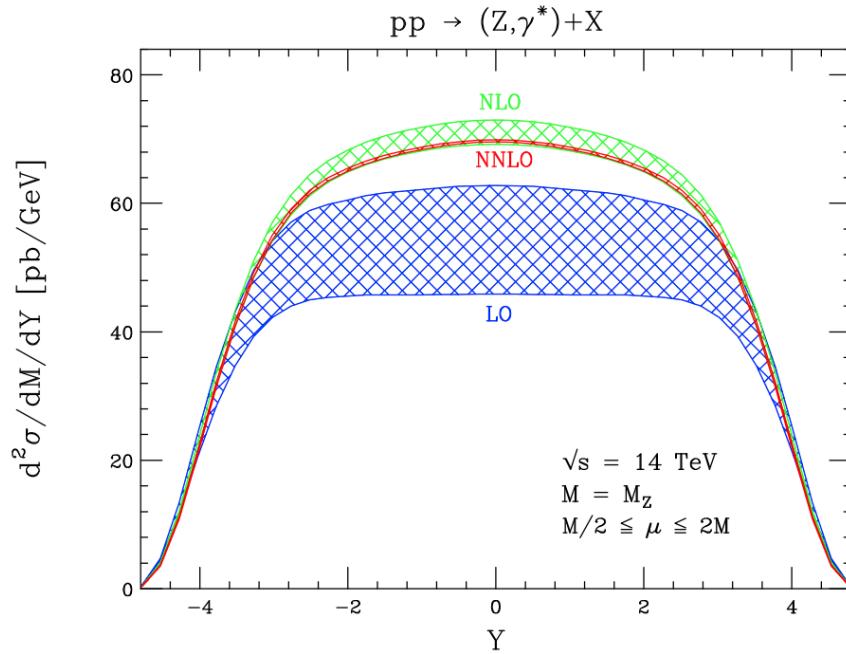
The importance of higher-order calculation is that it allows too describe more processes, as can be shown with the following example. In a  $Z$  boson production:

- 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 annihilation of a quark and an antiquark at LO (a), NLO (b), 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 a NLO calculation is the reduction of the dependence of calculation on the unphysical renormalization ( $\mu_R$ ) and factorization ( $\mu_F$ ) scales. It is proven that higher order calculations of observables calculated to order  $\alpha_s^n$  are dependent upon the unphysical scales only at order higher than  $\alpha_s^{n+1}$  [8]. 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 a increase in the cross section prediction and a reduction on 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 [8], section 6.

## 2.3 Parton Showers

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.

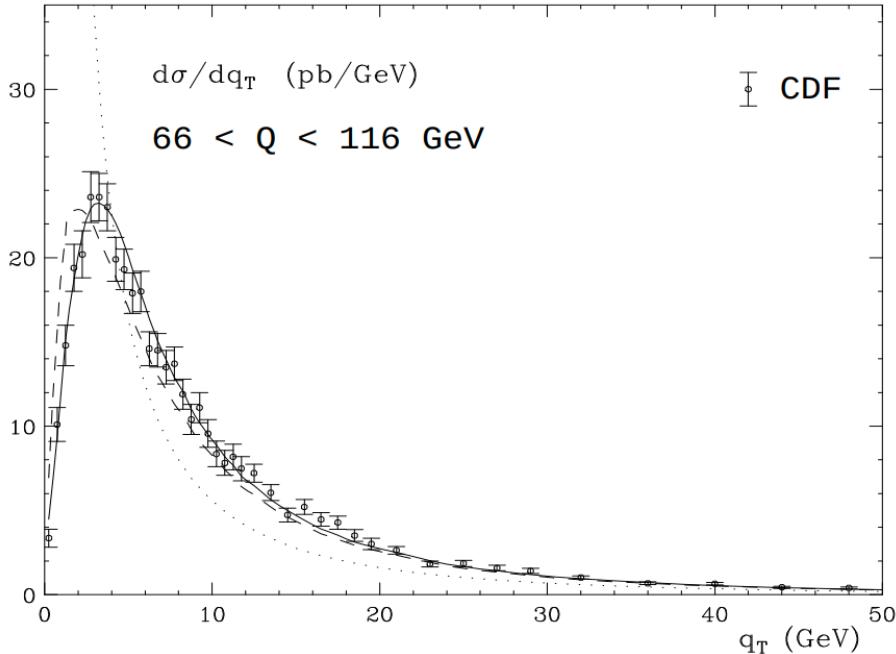
Different all-order approaches exist such as resummation techniques and parton showers. Resummation is based on the observation that in many quantities the smallness of the expansion coefficients  $\alpha_s$  is violated by large logarithmic enhancements. This takes the dominant contribution from each order and "resums" them by means of an evolution equation. The main problem in QCD is related to the fact that lot of quantities have corrections of the form  $\alpha_s^n \log^k(Q_i/Q_j)$  where  $Q_i$  and  $Q_j$  are two different energies scales, for example:

- Renormalization  $\mu_R = Q^2$  and factorization scales logs:  $\alpha_s^n \log^n(Q^2/\mu_f)$

Various methods to perform this resummation exist.

An example is the  $Z$  production  $p_T$  spectrum shown in Fig. 2.6: here the comparison between experimental CDF data and theoretical predictions is shown: in the low  $p_T$  region the *all-order* approach regularizes the divergence of the fixed order calculation and describes the data better.

An other *all-order* approach is based on the simulation of the so-called "parton showers". It is implemented in different Monte Carlo programs, as PYTHIA [18], HERWIG [19] and SHERPA [20]. The algorithm starts from few partons arising from



**Figure 2.6:** CDF data on  $Z$  production cross section at Tevatron collider, CDF experiment, the predictions from fixed order calculation (dotted) with resummation (dashed), and with the inclusion of power corrections (solid) are compared. Figure take from [17]

a hard interaction. Then, as the energy scale at which we examines the scattering decrease these harder partons can split, and so more partons are produced in the shower. The higher scale parton are related to the lower energy scale (close to  $\Lambda_{QCD}$ ) 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.

In the parton showering process, in addition to the kinematic variables (momentum fraction  $z$  and azimuthal angle  $\phi$ ) 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:

$$P_{q \rightarrow qg}(z) = C_F \frac{1+z^2}{1-z} ; \quad (2.9)$$

$$P_{g \rightarrow gg}(z) = C_A \frac{(1-z)(1-z)^2}{z(1-z)} ; \quad (2.10)$$

$$P_{q \rightarrow q\bar{q}}(z) = T_R(z^2 + (1-z)^2) ; \quad (2.11)$$

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 consist of two components: the initial-state radiation (ISR)

describing the emission from the incoming partons and the final-state radiation (FSR) describing emission of outgoing partons. Both ISR and FSR algorithms are based on these splitting kernels. The respective probabilities of emitting radiation as one moves in the decreasing evolution variable sequence are:

$$FSR : \quad \frac{d\mathcal{P}_{FSR}}{dp_T^2} = \frac{1}{p_T^2} \int \frac{dz}{z} \frac{\alpha_s}{2\pi} P(z) \quad ; \quad (2.12)$$

$$ISR : \quad \frac{d\mathcal{P}_{ISR}}{dp_T^2} = \frac{1}{p_T^2} \int \frac{dz}{z} \frac{\alpha_s}{2\pi} P(z) \frac{f'(x/z, p_T^2)}{f(x, p_T^2)} \quad . \quad (2.13)$$

We can write-out our Sudakov form factor by using the two probability in Eq. 2.12 and Eq. 2.13, as

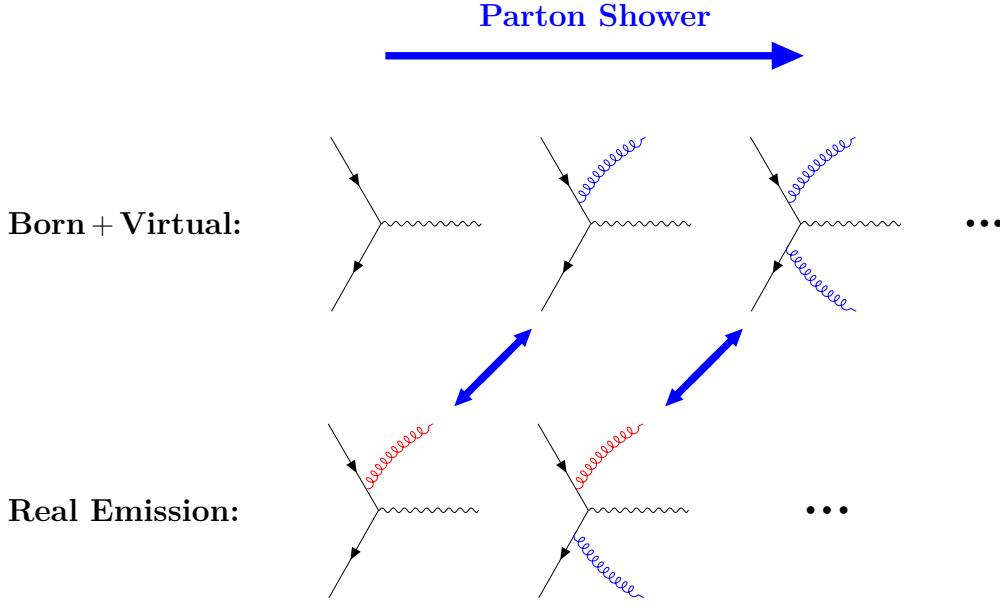
$$\Delta(p_T^2) = \exp \left( - \int_{p_{T0}}^{p'_T} \frac{d\mathcal{P}_{PS}}{dp_T^2} dp_T \right) \quad \text{with } PS = ISR, FSR \quad . \quad (2.14)$$

The Sudakov form factor give 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 resums all the effects from the soft and collinear gluon emission. For more details and some plots of different Sudakov form factor values see section 3.5 of [8].

### 2.3.1 Merging parton showers and 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 and widely separated are well described by matrix element calculations. So, the best approach would be to combine the two different descriptions, this would require an universal formalism for parton showers and matrix element calculations. This universal formalism was created in 2001 and it is called "Les Houches Accord" [21]. In order to combine the two approaches some care must be taken: there is the risk of double counting. Different techniques prevent this risk: for example CKKW [22] is used to combine LO matrix element calculations and parton shower.

A better way is to combine NLO matrix element calculation with parton showers: this is done by the FxFx margin scheme, developed by Frixione, Nason, Webber in the MC@NLO framework [23, 24, 25, 26]. 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. 2.7 by the red gluon line, 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. 2.7, the double counting sources are indicated by the blue arrows.



**Figure 2.7:** The FxFx margin scheme have to avoid this double counting. Feynman diagrams that can lead to a double counting are grouped with the violet arrow, the blue emission are related to the parton shower while the red ones to the NLO process.

## 2.4 Parton distribution functions

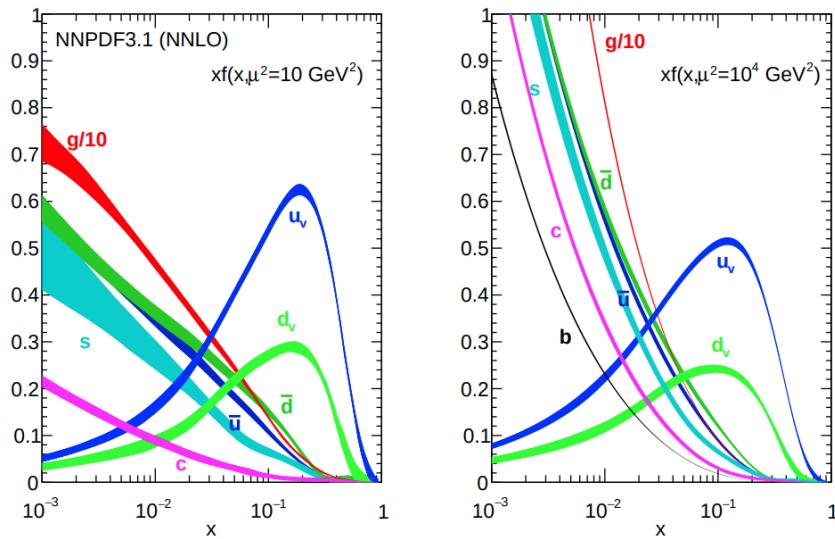
The last ingredient in our recipe is the knowledge of the quark and gluon distributions inside the two hadrons that undergo the scattering. We have already seen that these quantities are depending on the virtuality ( $Q^2$ ) of the interaction.

The information on the quark distribution inside a hadron  $f_{q/p}(x, Q^2)$  arises from lepton-hadron DIS experiments, from lepton-pair production in hadron-hadron collisions (Drell-Yan processes) and jet measurements to study gluon distribution  $f_{g/p}(x, Q^2)$ . All these quantities are the experimental input in order to evaluate the PDF inside the hadron while the  $Q$ -evolution is described by DGLAP equation. The evolution of the PDF can be run either with a NLO or with a NNLO calculations.

The kinematic region covered by experiments is shown in Fig. 2.2. At very low  $x$  and  $Q^2$  the DGLAP evolution is believed to be no longer applicable and a BFKL (Balitsky-Fadin-Kuraev-Lipatov) [27, 28] description must be used.

A lot of processes are available for the PDFs evaluation and a lot of PDF set have been generated, as an example Fig. 2.8 shows the NNPDF3.1 set [29] at NNLO for a virtuality  $Q^2 = 10 \text{ GeV}^2$  (left) and  $Q^2 = 10^4 \text{ GeV}^2$  (right). Note that the gluon contribution have been scaled of a factor 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. 2.8 we can also see that with increasing virtuality ( $Q^2$ ) at low  $x$  the density of the sea quarks increases: this is related to the fact that our hadrons are probed



**Figure 2.8:** The NNPDF3.1 NNLO PDFs set, evaluated at  $Q^2 = 10 \text{ GeV}^2$  (left) and  $Q^2 = 10^4 \text{ GeV}^2$  (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 our proton is probed to higher  $Q^2$  the resolution increase (higher  $Q$  correspond to smaller distance resolution) and so we have a bigger contribution from the sea quarks at low  $x$  values.

at higher energy and the probe resolution is proportional to the energy.

$$\text{Resolution} \sim \frac{\hbar}{Q} . \quad (2.15)$$

So, when probed at higher energy, the hadrons appear denser<sup>1</sup> than when are probed at lower energy. This, as will be discussed in the next chapter, is related to the higher number of interaction between partons in a single hadron-hadron collision. The phenomenon of having more than one interaction between partons in a single hadron-hadron collision is called *multiple parton interaction*: this concept will be discussed in the next chapter.

## 2.5 A real proton-proton collision

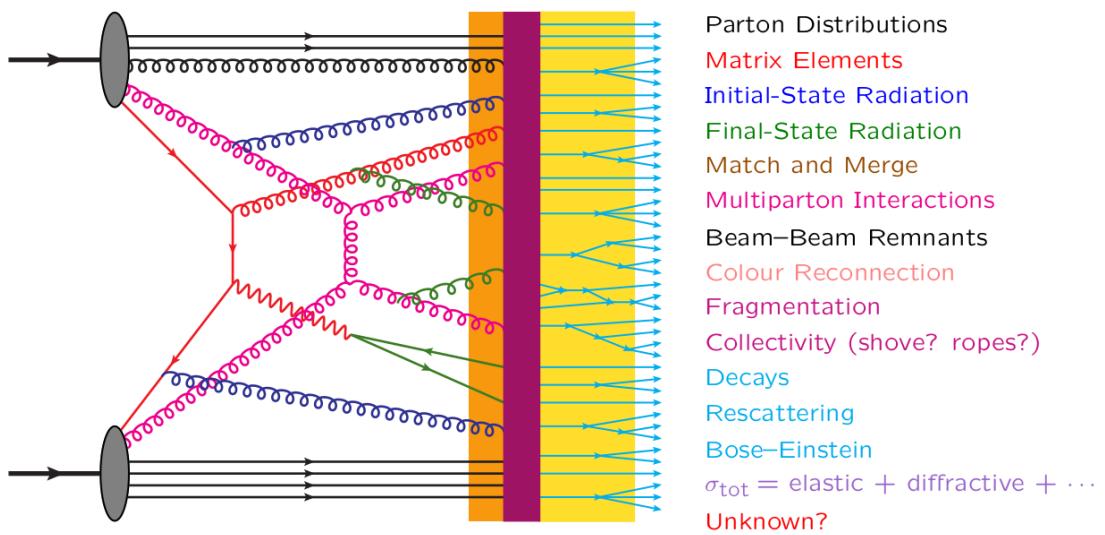
We have understood that the complexity in the description of a proton-proton collision arises from composite nature of the protons. In this chapter we discussed the importance of the QCD factorization theorem that help us in the calculation of the hadronic cross section with the convolution between the partonic cross section and the parton distribution functions (PDF). We have discussed the importance of the parton shower algorithm where a set of partons are evolved in a more complex final state by emissions in the initial and final states.

All these processes are important in the description of a real proton-proton collision

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

but also the partons that are left unscattered are non-color singlet and can contribute to the complex final state observed in the experiments, and additionally, as mentioned before, nothing prevent additional partons scatterings from taking place and increasing more and more the complexity of the partons final state.

Another problem is related to the not-well-understood hadronization process of all the final-state partons involved in the hadron-hadron interaction. Hadronization is not known from first principles and different models have been implemented in different programs: the *cluster fragmentation model* implemented in HERWIG and the *string fragmentation model* in PYTHIA All these processes are schematically shown in Fig. 2.9.



**Figure 2.9:** A schematic representation for a  $pp$  collision. Reading the image from left to right one can have an idea on the evolution of the system. The two incoming hadrons enter the scattering from the left side, the red line indicate 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. Than the new formed hadrons (lightblue) can undergo to different decays.

Next chapter will describe in more details the PYTHIA Monte Carlo generator, that is currently used to simulate many processes in proton-proton collision at the LHC. PYTHIA introduces different free parameters that need to be tuned with experimental data (from Tevatron and LHC). The tune methods are described in Chapter 5 along with the description of some already existing tune for the underlying event in proton-proton collision.

# Chapter 3

## Multiple Parton Interactions

By the fact that the hadrons are viewed as "bunches" of partons, it is likely that in the same hadron-hadron collision different couples of partons can undergo to a scattering. This phenomenon is known as *Multiple Parton Interactions* (MPI) and it 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 we are going to tune in the following sections.

### 3.1 Basic Concepts

The main hypothesis of the multiple interactions models is that: the QCD factorization theorem is true not only for the hard process but also for the other partons scatters.

So, we can write:

$$\frac{d\sigma_{\text{int}}}{dp_\perp} = \sum_{i,j,k,l} \int dx_1 \int dx_2 \int d\hat{t} f_i(x_1, Q^2) f_j(x_2, Q^2) \frac{d\hat{\sigma}_{ij \rightarrow kl}}{d\hat{t}} \delta\left(p_\perp^2 - \frac{\hat{t}\hat{u}}{\hat{s}}\right) . \quad (3.1)$$

This represent the interaction differential cross section for the hadron-hadron collisions, where  $\frac{d\hat{\sigma}_{ij \rightarrow kl}}{d\hat{t}}$  is the differential cross section for QCD hard  $2 \rightarrow 2$  processes, this processes are the one reported in Table 3.1.

In Eq. 3.1 we used the Mandelstam variables associated to the partonic system:

$$\hat{s} = (p_1 + p_2)^2 = (p_3 + p_4)^2 = x_1 x_2 s \quad (3.2)$$

$$\hat{t} = (p_1 - p_3)^2 = (p_2 - p_4)^2 \quad (3.3)$$

$$\hat{u} = (p_1 - p_4)^2 = (p_2 - p_3)^2 \quad (3.4)$$

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.

We assume also that the "hardness" of processes is defined by the  $p_T$  scale of the interaction ( $Q^2 = p_T^2$ ).

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

**Table 3.1:** Parton-Parton differential cross sections ( $2 \rightarrow 2$  QCD process), can be calculated in pQCD evaluating the matrix element for each process involving quark, antiquark and gluon. Table from [30]

As you can see from the formulae in Table 3.1 at small scattering angles, for  $t \rightarrow 0$ , the t-channel gluon exchange processes  $qq' \rightarrow qq'$ ,  $qg \rightarrow qg$  and  $gg \rightarrow gg$  dominate the full matrix element. For scatterings that are soft relative to  $\hat{s}$ ,  $|\hat{t}| \ll \hat{s}$ , we can approximate  $|\hat{t}|$  as:

$$p_T^2 = \frac{\hat{t}\hat{u}}{\hat{s}} = \frac{\hat{t}(-\hat{s}-\hat{t})}{\hat{s}} \approx |\hat{t}| \quad , \quad (3.5)$$

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

$$\hat{\sigma}_{gg} : \hat{\sigma}_{qg} : \hat{\sigma}_{qq} = \frac{9}{4} : 1 : \frac{4}{9} \quad . \quad (3.6)$$

So, the Eq. 3.1 can be rewritten like

$$\frac{d\sigma_{int}}{dp_T^2} \approx \int \int \frac{dx_1}{x_1} \frac{dx_2}{x_2} F(x_1, p_T^2) F(x_2, p_T^2) \frac{d\hat{\sigma}_{2 \rightarrow 2}}{dp_T^2} \quad , \quad (3.7)$$

whit:

$$\frac{d\hat{\sigma}_{2 \rightarrow 2}}{dp_T^2} = \frac{8\pi\alpha_s^2(p_T^2)}{9p_T^4} \quad ; \quad (3.8)$$

$$F(x, Q^2) = \sum_q (x q(x, Q^2) + x \bar{q}(x, Q^2)) + \frac{9}{4} x g(x, Q^2) \quad . \quad (3.9)$$

Now, we can integrate the Eq. 3.7:

$$\sigma_{int}(p_{T min}) = \int_{p_{T min}^2}^{(\sqrt{s}/2)^2} \frac{d\hat{\sigma}_{2 \rightarrow 2}}{dp_T^2} dp_T^2 \propto \frac{1}{p_{T min}^2} \xrightarrow{p_{T min} \rightarrow 0} \infty \quad (3.10)$$

The total cross section is divergent in the limit  $p_T \rightarrow 0$ , this divergence is shown in Fig. 3.1. Due to this divergence the total interaction cross section at some  $p_T$  scale can exceed the total proton-proton cross section.

To understand this paradox should be noted that the interaction cross section described in Eq. 3.10 is related to the interaction probability between two partons and counts the number of interactions, while the total proton-proton cross section  $\sigma_{pp}$  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.

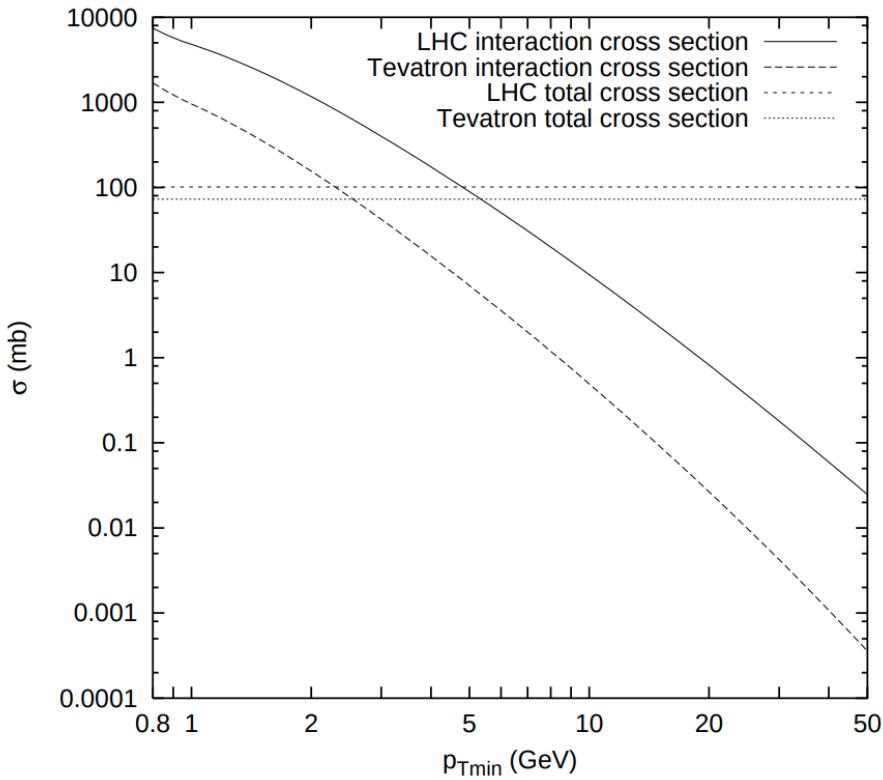
So, the ratio between this two quantities is perfectly allowed to be larger than unity, we can write it down as:

$$\frac{\sigma_{int}(p_{T min})}{\sigma_{tot}} = \langle n \rangle(p_{T min}) \quad . \quad (3.11)$$

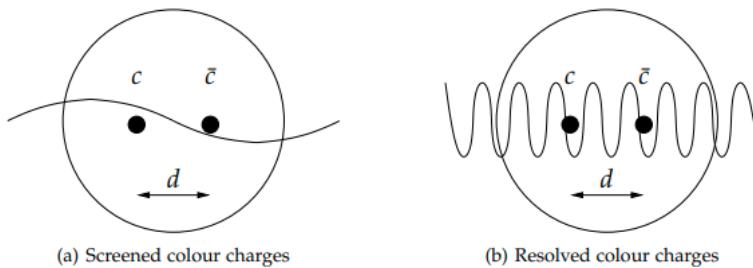
Furthermore, we have to consider the *screening effect*: in fact the incoming hadrons are color singlet objects. Therefore, when the  $p_T$  of an exchanged gluon is small, and so the associated wavelength large, this gluon cannot longer resolve the color charges and the effective coupling is decreased, this screening set a cutoff in the divergence. The screening effect is schematically shown in Fig. 3.2.

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

$$\frac{\alpha_s^2(p_{T0}^2 + p_T^2)}{\alpha_s^2(p_T^2)} \frac{p_T^4}{(p_{T0}^2 + p_T^2)^2} \quad . \quad (3.12)$$



**Figure 3.1:** This figure shows the interaction cross section ( $\sigma_{\text{int}}$ ) at Tevatron ( $p\bar{p}$ ,  $\sqrt{s} = 1.8 \text{ TeV}$ ) and at LHC ( $pp$ ,  $\sqrt{s} = 14 \text{ TeV}$ ). The flat lines are the corresponding values for the total cross section. The interaction cross section that arise from Eq. 3.10 is divergent for  $p_{T\text{min}} \rightarrow 0$  in reality a dumping of this divergence is expected due to the color screening effect.



**Figure 3.2:** A picture of the screening effect. The left figure show two color charge that are not resolved by the gluon in fact the wave length is greater than the spatial separation,  $d$ , of the two charges. So resolution of the probe is not enough to discriminate the color charges. While on the right figure the two charges are very well distinct.

This factor contains the phenomenological regularization of the divergence, with the factor  $p_{T0}$  that have to be tuned to data.

Now the interaction cross section is smoothly regularized and therefore finite.

To be notice that: parameter  $p_{T0}$  do not have to be energy-independent, but since the energy is related with the sensibility of our probe, higher energy is related with the capacity of probing PDFs to lower  $x$  values (as discussed previously, see Fig. 2.8)

and in this low  $x$  region the number of partons rapidly increases. So, the partons are closer packed in this regions 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  $p_{T0}$  respect to the center-of-mass energy

$$p_{T0}(\sqrt{s}) = p_{T0}^{ref} \left( \frac{\sqrt{s}}{E_{CM}^{ref}} \right)^{E_{CM}^{pow}} . \quad (3.13)$$

In the next section we are going to discuss how this and other effects fit into the Monte Carlo generator `pythia8`.

## 3.2 Pythia8 Monte Carlo events generator

`Pythia8` is a standard tool for the simulation of events in high energy collisions. `Pythia` contains the evolution from a few-body system to a complex multiparticles final state.

### 3.2.1 Parton Shower

In `Pythia8` all the contributions from Initial State Radiation (ISR), Final State Radiation (FSR) and Multi Parton Interactions (MPI) are interleaved into a single common sequence of decreasing  $p_T$ .

The parton shower has been described in the previously section. The solution to the DGLAP equation is given putting Eq. 2.12 and Eq. 2.13 in Eq. 2.14 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:

$$p_T^2 = \begin{cases} (1-z)Q^2 & ISR \\ z(1-z)Q^2 & FSR \end{cases} , \quad (3.14)$$

so, in the two cases: for the FSR  $Q_{FSR}^2 = (p^2 - m_0^2)$  in fact a time-like virtuality ( $p^2 > 0$ ) is implicated, while for the ISR we have  $Q_{ISR}^2 = (-p^2 + m_0^2)$  with a space-like virtuality ( $p^2 < 0$ ). In both case  $Q^2$  values are positive defined. Than the actual strong of the radiation is set by the choices of  $\alpha_s_{ISR}$  and  $\alpha_s_{FSR}$  values.

### 3.2.2 Multiple Parton Interactions in Pythia

The MPI, as said before, are also generated in a decreasing  $p_T$  sequence. So the hardest MPI is generated first. Than we can write the probability for an interaction,  $i$ , to occur at a scale  $p_T$  using a Sudakov-type expression (as we have done for the ISR and the FSR):

$$\frac{d\mathcal{P}_{MPI}}{dp_T} = \frac{1}{\sigma_{ND}} \frac{d\sigma_{2 \rightarrow 2}}{dp_T} \exp \left( - \int_{p_T}^{p_T^{i-1}} \frac{1}{\sigma_{ND}} \frac{d\sigma_{2 \rightarrow 2}}{dp'_T} dp'_T \right) . \quad (3.15)$$

### 3.2.3 Momentum and flavour conservation

One problem is to achieve momentum conservation, so we need to take into account the modification in the PDF by the  $i - 1$  interaction. To do that in **Pythia** the PDF are rescaled to the remaining available  $x$  range, adjusting their normalization.

We need to take into account the momentum fraction  $x_i$  removed from the hadron remnant by the  $i - th$  interaction. This is done evaluating PDF not at  $x_i$  but at a rescaled value

$$x'_i = \frac{x_i}{X} \quad \text{with} \quad X = 1 - \sum_{j=1}^{i-1} x_j \quad . \quad (3.16)$$

So, using these quantities, we can rewrite our PDFs as:

$$f_i(x, Q^2) \longrightarrow \frac{1}{X} f_0\left(\frac{x}{X}\right) \quad , \quad (3.17)$$

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. We have now the full forms of the PDFs used for the  $i - th$  MPI:

$$f_i(x, Q^2) = \frac{N_{fv}}{N_{fv0}} \frac{1}{X} f_{v0}\left(\frac{x}{X}, Q^2\right) + \frac{a}{X} f_{s0}\left(\frac{x}{X}, Q^2\right) + \sum_j \frac{1}{X} f_{cj0}\left(\frac{x}{X}, Q^2\right) \quad , \quad (3.18)$$

$$g_i(x, Q^2) = \frac{a}{X} g_0\left(\frac{x}{X}, Q^2\right) \quad , \quad (3.19)$$

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 we have  $N_u = 2$ ,  $N_d = 1$ );
- $f_{s0}$  the sea-quark PDF;
- $f_{cj}$  the companion PDF, this arise from the splitting  $g \rightarrow q\bar{q}$  and a quark  $j$  is kicked out.

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

### 3.2.4 Impact Parameter Dependence

The simplest hypothesis for the multiple interaction simulation, it is to assume the same initial state for all hadron collisions without dependencies on the impact parameter.

The more realistic scenario it is to include the possibility that each collision could be characterized by a different impact parameter  $b$ , where a small  $b$  value correspond 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 assumption 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 coiche is a double Gaussian:

$$\rho(r) \propto \frac{1-\beta}{a_1^3} \exp\left\{-\frac{r^2}{a_1^2}\right\} + \frac{\beta}{a_2^3} \exp\left\{-\frac{r^2}{a_2^2}\right\}, \quad (3.20)$$

where a fraction  $\beta$  of matter is contained in a radius  $a_2$ , and the rest in a larger radius  $a_1$ .

Now, for a given matter distribution  $\rho(r)$ , the time-integrated overlap function of the incoming hadrons during the collision is given by:

$$\mathcal{O}(b) = \int dt \int d^3x \rho(x, y, z) \rho(x + b, y, z + t). \quad (3.21)$$

Assuming the matter distribution function in Eq. 3.20 and inserting it into Eq. 3.21 we get the following expression.

$$\mathcal{O}(b) \propto \frac{(1-\beta)^2}{2a_1^2} \exp\left\{-\frac{b^2}{2a_1^2}\right\} + \frac{2\beta(1-\beta)}{a_1^2 + a_2^2} \exp\left\{-\frac{b^2}{a_1^2 + a_2^2}\right\} + \frac{\beta^2}{2a_2^2} \exp\left\{-\frac{b^2}{2a_2^2}\right\} \quad (3.22)$$

this 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  $\langle \tilde{n} \rangle \propto \mathcal{O}(b)$ .

So, these assumption change the so-far Poissonian nature of the framework<sup>1</sup>

$$\mathcal{P}(\tilde{n}) = \langle \tilde{n} \rangle^{\tilde{n}} \frac{e^{-\langle \tilde{n} \rangle}}{\tilde{n}!} \quad (3.23)$$

Now the request that at least one parton interactions in the hadron-hadron collision, ensures that we get a finite total cross section. So the probability that an event is produced by two hadrons scattering with impact parameter  $b$ :

$$\mathcal{P}_{\text{int}} = \sum_{\tilde{n}=1}^{\infty} \mathcal{P}_{\tilde{n}(b)} = 1 - \mathcal{P}_0 = 1 - e^{-\langle \tilde{n}(b) \rangle} = 1 - e^{-k\mathcal{O}(b)} \quad (3.24)$$

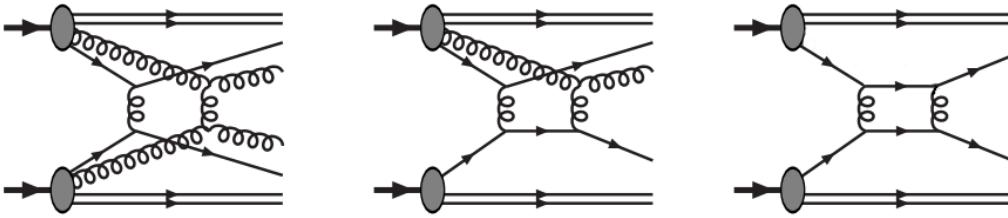
So we have now that the number of interaction per event is give by (for impact parameter  $b$ ):

$$\langle n(b) \rangle = \frac{\langle \tilde{n}(b) \rangle}{\mathcal{P}_{\text{int}}} \quad (3.25)$$

so, this have modified the probability distribution of interactions number from a Poissonian to a narrower one at each  $b$  fixed.

### 3.2.5 Parton rescattering

It is not necessary that the partons undergoing to a MPI are a different partons couple from the one scattered before. As shown in Fig. 3.3 MPI can also arise when a parton scatters more than once against partons from the other beam, this is call



**Figure 3.3:** This figure shows parton rescattering. On the right image a double  $2 \rightarrow 2$  scattering (no rescattering) is shown; on the middle a parton rescattering process where only one of the rescattered partons have already scattered; on the right image a parton rescattering with both the partons that undergo to rescattering have already been scattered.

*parton rescattering.*

We can have 3 type of MPI:

1. No ones of the partons that enter in the second scattering undergoes to another scatter before (Fig. 3.3 left);
2. Only one of the two partons has already been scattered (Fig. 3.3 middle);
3. Both the partons have already been scattered before (Fig. 3.3 right).

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

The simulation of parton rescatters start from the evaluation of the parton density as:

$$f(x, Q^2) \longrightarrow f_{\text{rescaled}}(x, Q^2) + \underbrace{\sum_n \delta(x - x_n)}_{\substack{\text{hadron remnant} \\ \text{scattered parton(s)}}}, \quad (3.26)$$

where the  $\delta(x - x_i)$  takes into account the already scattered partons that have a fixed momentum fraction  $x_n$ . While the hadron remnant is still described by a continuous momentum density, scaled to achieve momentum conservation:

$$\int_0^1 \left( f_{\text{rescaled}}(x, Q^2) + \sum_n \delta(x - x_n) \right) dx = 1. \quad (3.27)$$

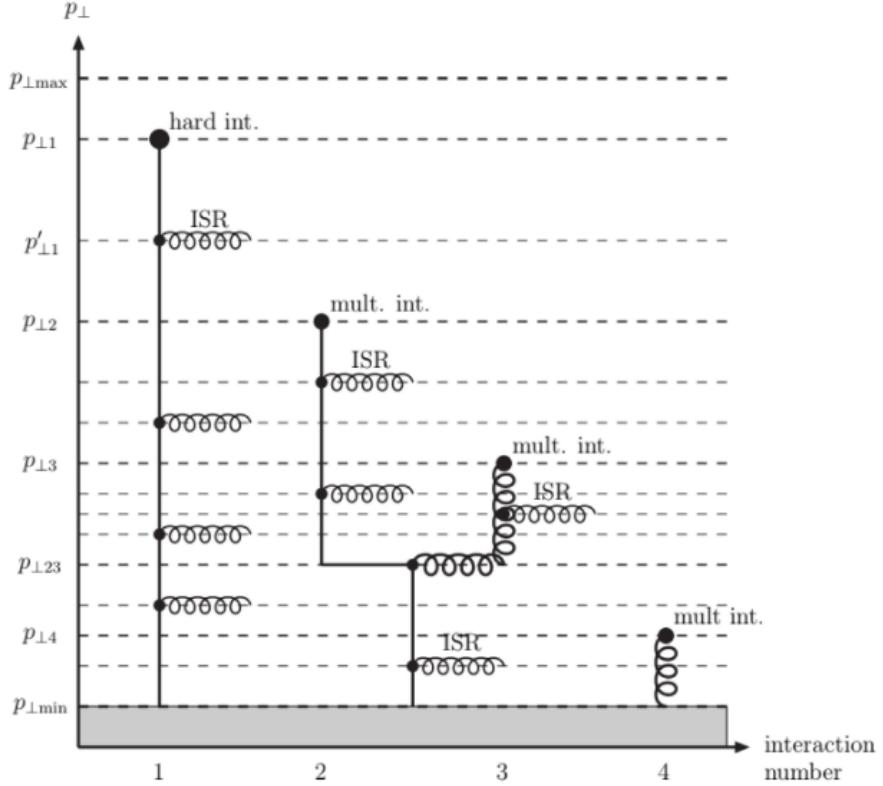
### 3.2.6 Interleaving of Multiple Interaction and Parton Shower

As discussed above the MPI are simulated in Pythia following a decreasing  $p_T$  evolution. So we have that the total probability is given from the composition of

<sup>1</sup>the Poissonian distribution, in Eq. 3.23 describe the probability of having  $n$  interactions at each impact parameter. If the matter distribution have an infinite tail (like our in Eq. 3.20) events may be obtained with arbitrarily large  $b$  values. This can be a problem for the definition of the total hadron-hadron cross section

the various contributions:

$$\frac{d\mathcal{P}}{dp_T} = \left( \frac{d\mathcal{P}_{MPI}}{dp_T} + \sum \frac{d\mathcal{P}_{ISR}}{dp_T} + \sum \frac{d\mathcal{P}_{FSR}}{dp_T} \right) \times \exp \left\{ - \int_{p_T}^{p_T^{i-1}} \left( \frac{d\mathcal{P}_{MPI}}{dp'_T} + \sum \frac{d\mathcal{P}_{ISR}}{dp'_T} + \sum \frac{d\mathcal{P}_{FSR}}{dp'_T} \right) dp'_T \right\} \quad (3.28)$$



**Figure 3.4:** 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. 3.4 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.

### 3.2.7 Beam Beam Remnants and primordial $k_T$

What is left after that the evolution is end?

The evolution in  $p_T$  can create an arbitrary complicate final state. This final state

contains contributions from the scattered and unscattered partons that don't enter the  $p_T$  evolution. The last are the so called *Beam Beam Remnants* (BBR). BBR take into account the number of valence quarks remaining and the number of sea quarks required for the overall flavour conservation.

To ensure momentum conservation the BBR have to take all the remaining longitudinal momentum that is not extracted by the MPI initiators.

### Primordial $k_T$

We have 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:

$$k_T \simeq \frac{\hbar}{r_p} \approx \frac{0.2 \text{ GeV} \cdot \text{fm}}{0.7 \text{ fm}} \approx 0.3 \text{ GeV} , \quad (3.29)$$

but, as an example, to reproduce the data for the  $p_T$  distributions of  $Z$  bosons produced in hadron–hadron collisions, one need a larger contribution. 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. 3.29.

In **Pythia** the primordial  $k_T$  is assigned to initiators sampling a Gaussian distributions for  $p_x$  and  $p_y$  independently with variable width  $\sigma$

$$\sigma(Q, \hat{m}) = \frac{Q_{1/2} \sigma_{\text{soft}} + Q \sigma_{\text{hard}}}{Q_{1/2} + Q} \frac{\hat{m}}{\hat{m}_{1/2} + \hat{m}} \quad (3.30)$$

Where  $Q$  is the hard-process renormalization scale for the main hard process and the  $p_T$  scale for subsequent MPI.  $\sigma_{\text{soft}}$ ,  $\sigma_{\text{hard}}$  are the minimum and maximum width,  $\hat{m}$  is the invariant mass, while  $Q_{1/2}$  and  $\hat{m}_{1/2}$  the halfway values between the two extremes.

### 3.2.8 Color Reconnection and Hadronization

The last important step at parton level is the *color reconnection*. Color reconnection 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  $N_c \rightarrow \infty$ . Now, moving to real case where  $N_c = 3$  all this 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 the reconnection is performed giving to each system a probability to reconnect given by:

$$\mathcal{P}_{\text{rec}} = \frac{p_{T \text{ rec}}^2}{(p_{T \text{ rec}}^2 + p_T^2)} \quad p_{T \text{ rec}} = R \times p_{T0} , \quad (3.31)$$

the `ColorReconnection:range R` is a user-tunable parameter while  $p_{T0}$  is the same parameter defined in MPI simulation.

With this definition for the probability, it is clear that system with low  $p_T$  are more likely reconnected to other. This idea find is justification in the fact that lower  $p_T$  values correspond to larger spatial extension and so these strings have more chance to overlap with other and so to reconnect.

## Hadronization

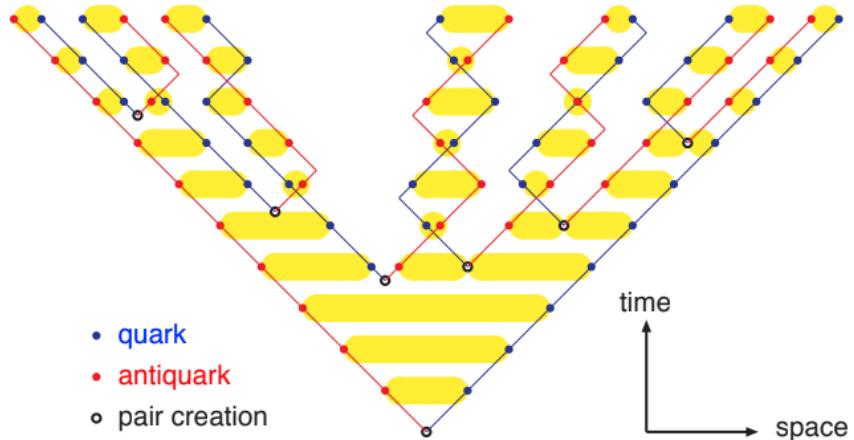
The hadronization takes all these partons (color strings) and transforms them in a set of color-neutral hadrons. Hadronization is based solely on the *Lund string fragmentation model* [31, 32].

Lund model basic idea is to break the color line and to reduce the total string length, the string is representative of the potential

$$V(r) = -\frac{a}{r} + \kappa r \quad \text{with } \kappa \approx 1 \text{ GeV/fm} , \quad (3.32)$$

where  $\kappa$  is the string tension. This potential is a combination of an attractive (Coulomb) potential ad a linear potential that phenomenologically include quarks confinement. 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. 3.5: The  $q\bar{q}$  system evolves in space increasing the string length at some point the distance is to large and is convenient to break the string into two strings.



**Figure 3.5:** The lund Model is schematically shown here. The string evolution in time is shown vertically while the spatial position horizontally. As the partons move apart at some point become 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 undergoes to hadron rescattering and decay. These are the hadrons that are revealed by the detectors.

### 3.3 PYTHIA summary

To summarize PYTHIA8 is able to simulate high energy hadron-hadron collisions. The evolution of the system is simulated in a common decreasing- $p_T$  sequence the master formula for the evolution is written in Eq. 3.28.

This evolution start from an hard scale that is the scale of the main parton hard scattering that is described by a fixed ME calculation, PYTHIA can be interfaced with external frameworks for the ME step as POWHEG and MAD-GRAPH5 AMC@NLO. Then the parton shower is started with the simulation of ISR, FSR and MPI also the parton rescattering is allowed. Once the evolution is ended the hadronization transform the partons in a set of final hadrons these hadron than decay and rescatters against each others before the detection.

All these processes are describe mainly by phenomenological model, 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 the previously sections) that have to be tuned with data to give to PYTHIA the ability of simulate real events.

In the next chapter we are going to focus on the study of the underlying event and so on all the soft events related to an hard scattering in an hadron-hadron collision.

# Chapter 4

## CMS Detector, Observable to Study the Underlying Event and Existing Tunes

The underlying events UE are all the processes not associated with the primary hard scatter in an hadron-hadron collision.

All the process described in the previous section: ISR, FSR, MPI, and BBR and their interactions with color exchanges, contribute to the Underlying Event (UE) in the 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  $l^-$ ) and in the end these two leptons observed by the experiment, we have that the hard scattering is represented from the scatters 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 complexes finals states originating from this scatters that contribute to the observables.

In this chapter these observables sensible to the UE are introduced and described with more detail.

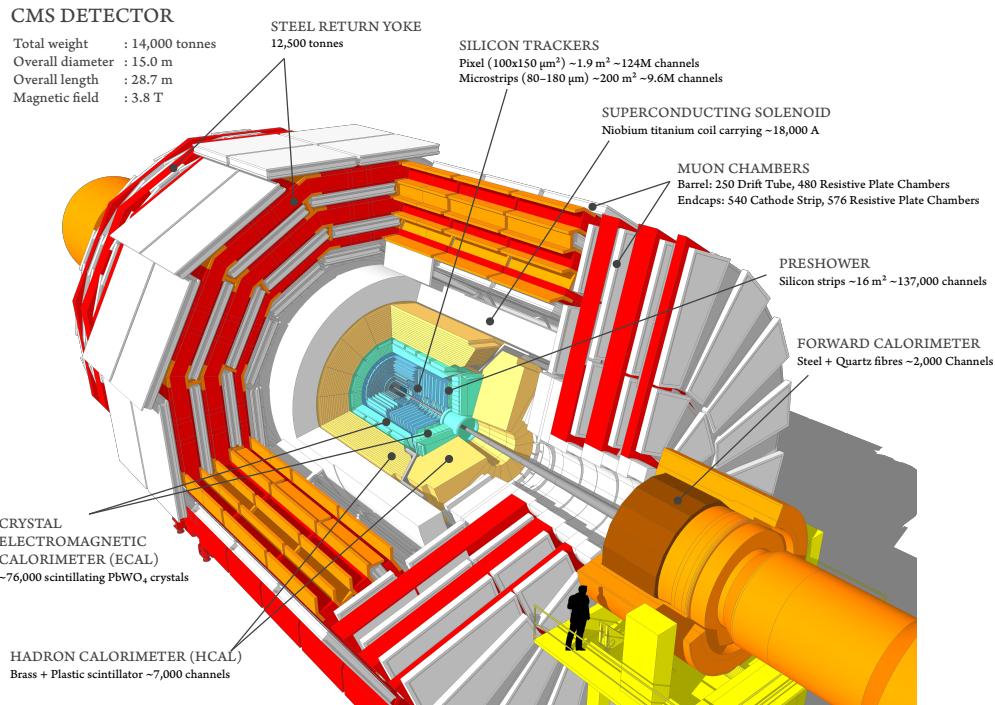
### 4.1 Compact Muon Solenoid Detector

In this section we are going to briefly 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 particle physics experiments in particular one is reserved for the CMS detector.

## CHAPTER 4. CMS DETECTOR, OBSERVABLE TO STUDY THE UNDERLYING EVENT AND EXISTING TUNES

---

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 ( $\text{PbWO}_4$ ) crystal electromagnetic calorimeter (ECAL), and hadron calorimeter (HCAL) composed by alternating layers of brass and scintillator material; both calorimeters are composed by a barrel, an endcap and a forward section. The outer region are covered by the muon detectors, these are composed from 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. 4.1.



**Figure 4.1:** An illustration of the CMS detector [33].

The silicon tracker is responsible of the measurement of charged particles tracks within the pseudorapidity range  $|\eta| < 2.5$ . The typically track resolutions, for non-isolated particles of  $1 < pT < 10\text{GeV}$  and  $|\eta| < 1.4$ , are 1.5% in  $p_T$ . This inner tracking system is the central detector for the charge particles tracks reconstruction and, thanks to the high magnetic field, for the momentum measurements in CMS. The ECAL barrel cover the pseudorapidity range  $|\eta| < 1.479$  it consist of about 61200  $\text{PbWO}_4$  crystal while the endcaps cover the pseudorapidity range  $1.479 < |\eta| < 3.0$ . This was build in order to have a resolution able to discriminate the two gamma in the Higgs boson decay  $H^0 \rightarrow \gamma\gamma$ .

The HCAL, instead, the barrel cover the pseudorapidity range  $|\eta| < 1.3$  while the endcaps  $1.3 < |\eta| < 3.0$  this detectors are really important in the hadronic jet 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 cover the pseudorapidity up to  $|\eta| < 2.4$

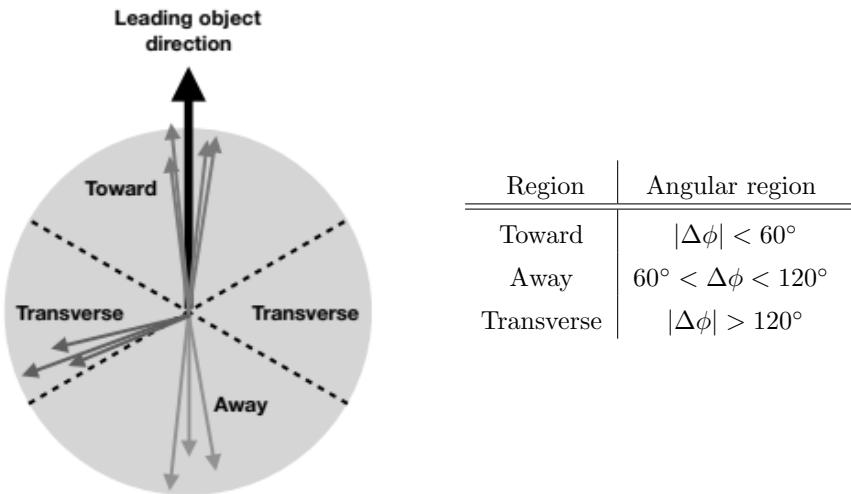
The LHC provide proton-proton collision at very high interaction rates, the beam crossing interval is 25 ns). So, it is impossible to store all the data, the rate is reduced in two steps: Level-1 (L1) Trigger and High-Level Trigger (HLT). This two step perform a first events selection keeping only the interest interactions.

In the next section we are going to discuss some of the distribution observed using CMS detector.

## 4.2 Minimum Bias Measurements and Underlying Event topology

A Minimum Bias (MB) measurement is a collection of inelastic events with a loose event selection. Which means the events are collected requiring the minimal interaction with the detector (the smallest possible bias). The most of the interactions in MB observation 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 an hadron-hadron collision is used. The analysis is performed on an *event-by-event* basis. In the analysis the direction of the leading object is used to define regions in the  $\eta - \phi$  space. Where  $\eta$  is the pseudorapidity defined as  $\eta = -\log \tan(\frac{\theta}{2})$ , while  $\phi$  is the azimuthal angle in the  $x - y$  plane. The last one is defined from the direction of the leading object as  $\Delta\phi = \phi - \phi_{\text{leading}}$ .



**Figure 4.2:** This figure shows the four regions for the description of the UE on a 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 in between the two other regions, these are the most important for the study of the UE.

The regions classification is shown in Fig. 4.2, we have:

- **Toward region:** the region that contains the leading object, this region contains the 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 mostly the particles produced by the hard interaction.
- **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, in fact in a dijets event the leading jet is expected in the toward region while the next-to-leading 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 regions includes both MPI and hard-process contamination. As an example, in events with 3-jets production these region can contains 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 depend on the type of event under observation and defer from one analysis to another. The charged-particle with largest  $p_T$  [34], the dilepton system in Drell-Yan observation [35, 36] or  $t\bar{t}$  events [37] can all be used as leading object in the analyses for the UE event.

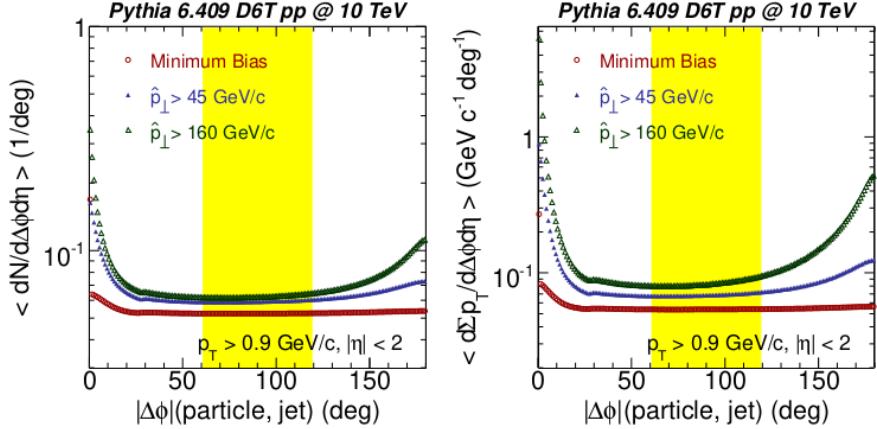
So we are interest in studying the transMAX and transMIN regions in particular the observable sensitive to UE in these regions. The main observables studied in these regions are: the charged-particle density and the charged-particle scalar- $p_T$  sum density in the  $\eta - \phi$  space.

In an hadron-hadron collision with two jets productions it is observed that not only the toward and away regions activity increases with the hardness of the collision ( $p_T^{max}$ ) but also the transverse regions activity increases with it. This is shown in Fig. 4.3, where the different color lines refer to different 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.

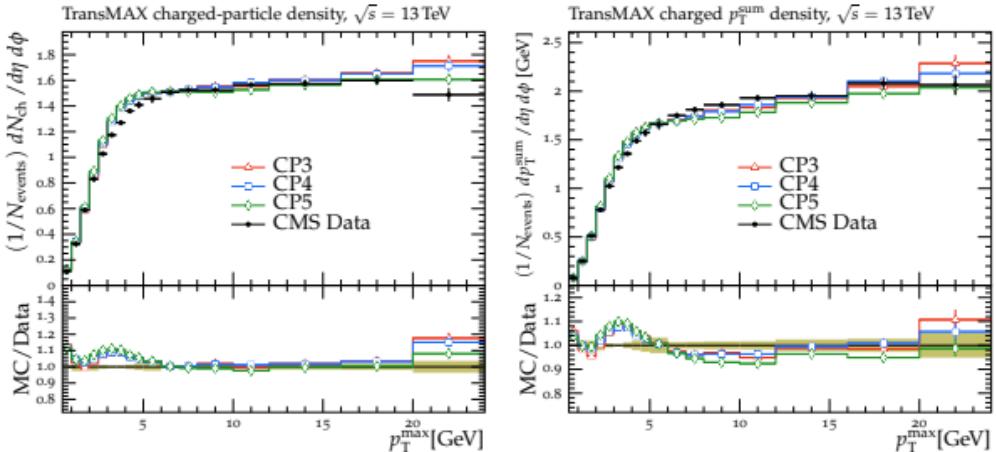
This increment in the transverse regions activity cannot be explained only by the increase of the FSR. To explain this rise one need to attribute activity in the transverse regions to MPI: the number of extra scatters increases with energy. This rise is related to the density functions for the partons inside the hadrons that increase when probed at higher energies and so the partons become denser packed in hadrons. In this way the extra scatters probability is larger to higher energies.

The evolution of these two quantities in 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. 4.4. The two distributions show a rapid raise at low leading-object  $p_T$  than a very low rise starts.

Now we want to look for the sensitivity of these observables to some PYTHIA8 parameters. In figure Fig. 4.5 is shown the effect of the MPI on booth the distributions shown before separately for TransMAX and TransMIN regions. The figure



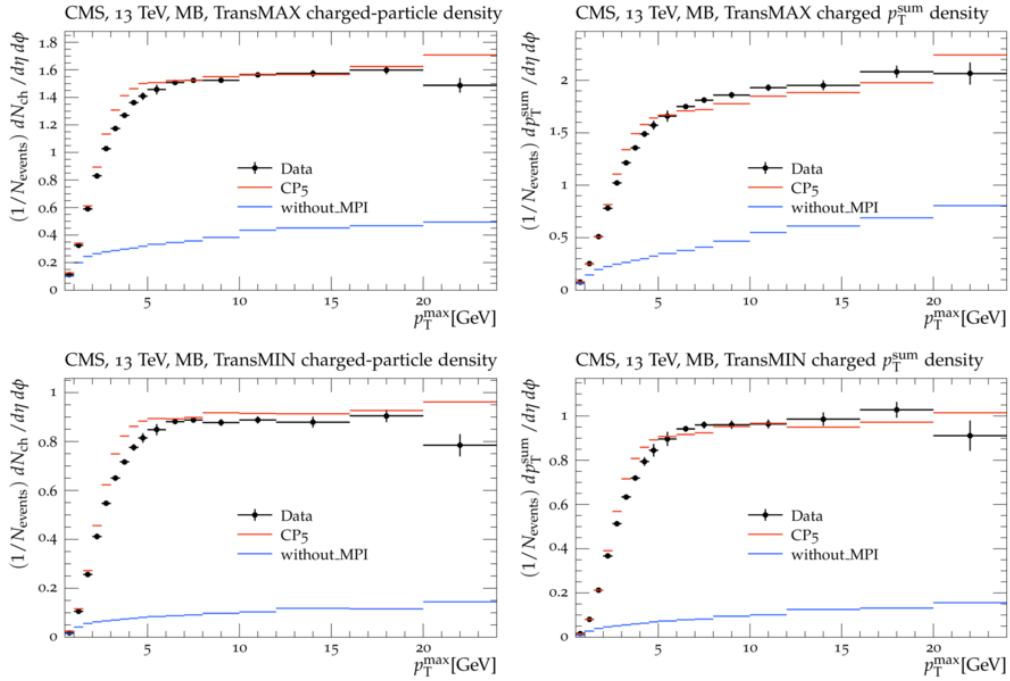
**Figure 4.3:** A comparison between three different scales for the interaction, in PYTHIA. The multiplicity of charged particles (left) and the scalar- $p_T$  sum of charged particles (right) are simulated. The activity in the transverse regions increase due two effects: the FSR this is related to the broadening of the away region 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 [38].



**Figure 4.4:** 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  $p_T$ ), the two distribution increase quite rapidly in the first bins than they saturate ( $\approx 6 - 7 \text{ GeV}$ ) 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 with more details in the next chapter. Figure from [39].

shows the case with MPI (red line) and the case when the MPI are switched off (blue line). As expected the MPI are needed to explain the activity in the transverse regions. When the MPI are switched-on the number of particle produced is higher than the case when they are off. From this figure is clear that MPI are the main contribution in the activity of these TransMAX and TransMIN regions. It's obviously that the discrepancy increases as the hard scale of the interaction increases, in fact in low  $p_T$  regions the  $p_T$  evolution start at a lower value so the phase-space,

for the extra collisions to occur, is smaller.



**Figure 4.5:** This image shows the effect of the MPI in the transMAX and TransMin regions. The contribution of the parton shower alone can't 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 [34].

Another important observation that was also pointed out in the Chapter 3.1 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. 4.6 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. 4.6 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. The data are also compared to existing different CMS tuning.

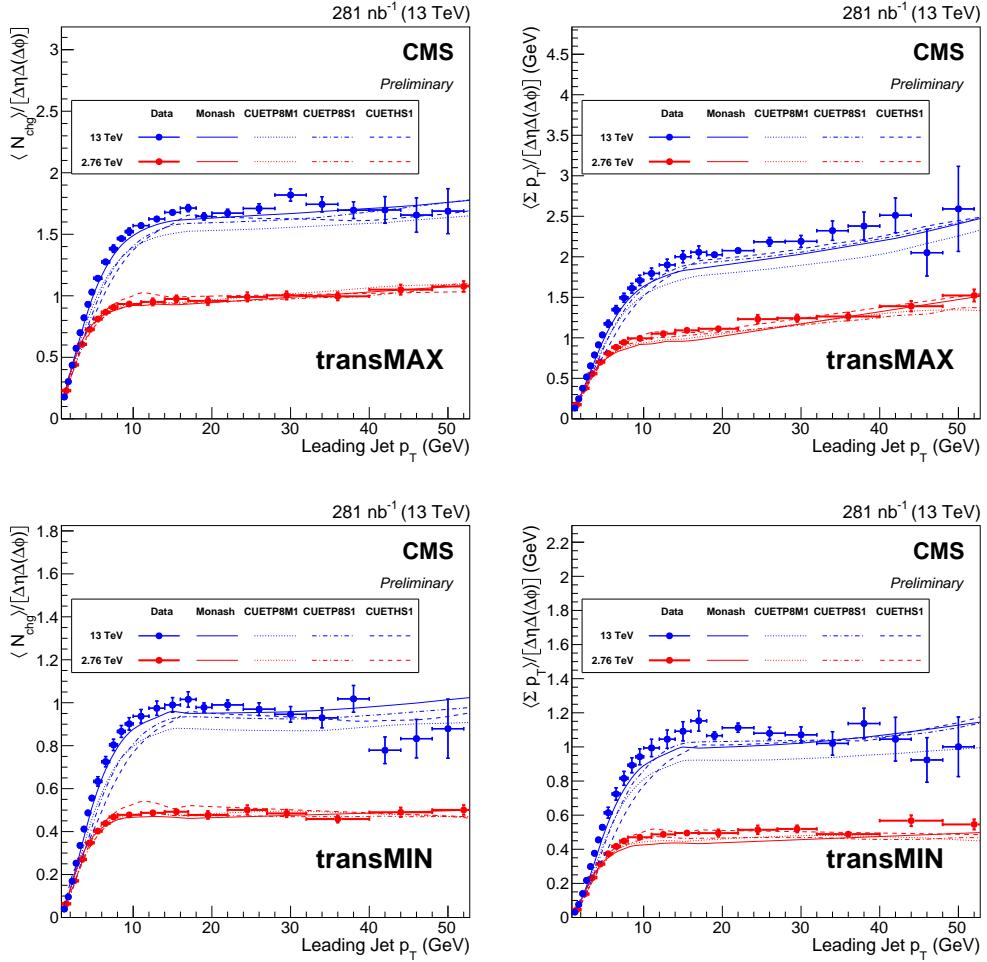
So these observables are the main ones we are going to use in our tune and used in previous tunes for the UE. The data from the activity in the TransMAX and TransMIN regions have been collected by a large number of experiment at different center-of-mass energies.

### 4.2.1 Observables in Z production processes

In the last part of this work we are going to tune the primordial  $k_T$  another important parameter in the simulation of hadron-hadron collisions. An unresolved question is that the primordial  $k_T$  value required for the description of the observed cross section for the Z production is very large respect to the expected value derived from the typical size of an hadron in Eq. 3.29. This very high value is not understood by first

## 4.2. MINIMUM BIAS MEASUREMENTS AND UNDERLYING EVENT TOPOLOGY

---



**Figure 4.6:** 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 [34]

principles so it is required a tune of this parameter in the Monte Carlo generator in order to describe the data in a good way.

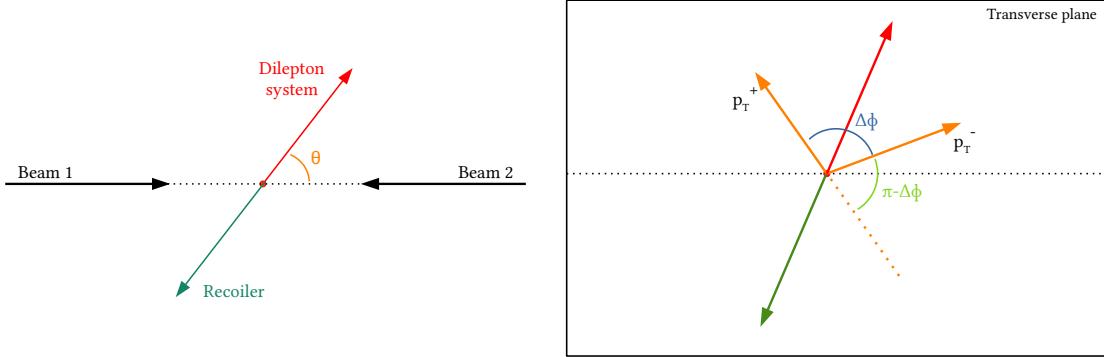
The observables to study the primordial  $k_T$  are the differential cross section for Z production in proton-proton collision in Drell-Yan observation as a function of the transverse momentum of the Z boson (Fig. 4.8 left) and as a function of the  $\phi_\eta^*$  angle (Fig. 4.8 right). Where the angle  $\phi_\eta^*$  is defined as:

$$\phi_\eta^* = \tan\left(\frac{\pi - \Delta\phi}{2}\right) \sin(\theta_\eta^*) \quad , \quad (4.1)$$

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

$$\cos(\theta_\eta^*) = \tanh\left(\frac{\eta^- - \eta^+}{2}\right) \quad , \quad (4.2)$$

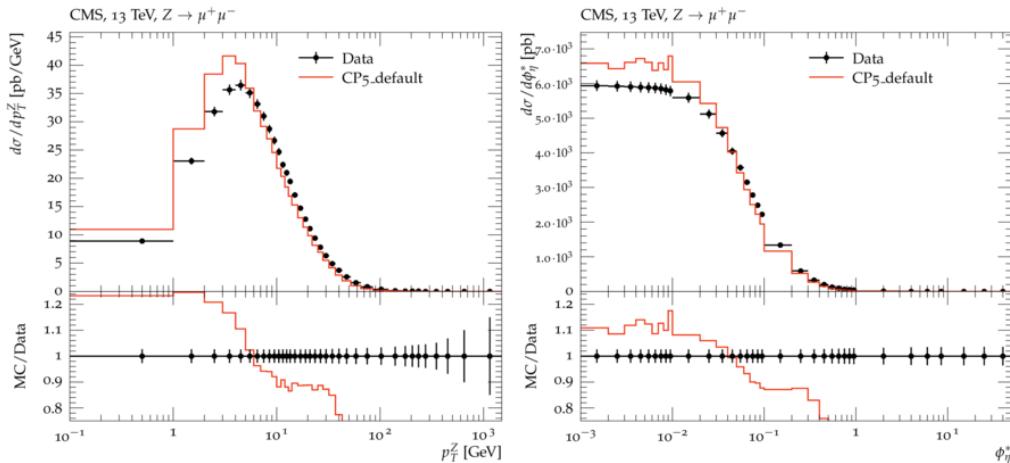
with  $\eta^+$  and  $\eta^-$  the pseudorapidities of the two leptons [40].



**Figure 4.7:** A schematic representation of a Drell-Yan observation with a description of the angles used in the definition of  $\phi_\eta^*$  in Eq. 4.1.

To full simulate the Z production spectrum we have to merge matrix element calculation and space shower to do this we used the FxFx merging scheme that have been introduced in Chapter 2. As shown in Fig. 4.8 the description with out FxFx is good only in the first bins. The strategy we follow for the tuning of the primordial  $k_T$  is to tune the Primordial  $k_T$  in this low region. From these two distribution we take only the first 5 bins each.

Than we are going to run the FxFx only for the comparison once the parameters are tuned. This is going to be discussed in Chapter 7 where we perform the primordial  $k_T$  tune.



**Figure 4.8:** Here are shown the distribution, from the [41] CMS analysis at  $\sqrt{s} = 13$  TeV, that we are going to use 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  $\phi_\eta^*$  angle. The red line is the 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.

In the following section we will introduce the existing CMS tunes for the minimum bias and UE. These tunes are able to reproduce very well the observables described in Section 4.2 at 1.96, 7 and 13 TeV. But as we are going to see in

Chapter 2 the tunes do not describe well the low region of the  $Z$  boson production cross section so a further tune on these observable is performed.

## 4.3 Previous Tune for the Underlying Event

In the paper [39] 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 changing the values of the coupling constant  $\alpha_s$  for the ISR, FSR, hard scattering and MPI and changing the order of the evolution for the  $\alpha_s$  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 a NLO one, while CP4 and CP5 tunes use a NNLO PDF set.

The tuned parameters are shown in Table 4.1 with the associated variation ranges and a recall on 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, $\epsilon$	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 4.1:** This table report the five parameters tuned for the UE and Minimum Bias in CP\* tunes, the variation ranges used for the sampling are shown in the last column. Table from [39]

These parameters are the ones that govern the number of MPI and the amount of Color Reconnections<sup>1</sup>. These CP\* tunes want to be a 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 tune: PROFESSOR.

### 4.3.1 The distributions used for the Tune

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

- The pseudorapidity distribution of charged hadrons ( $p_T, K_T, \pi$ ) measured for an inclusive selection in inelastic proton-proton collisions [42];
- Charged particle density and charged particle scalar  $p_T^{sum}$  in TransMIN and TransMAX regions at different  $\sqrt{s}$  (1.96 TeV [43], 7 TeV [44], 13 TeV [34]);
- The pseudorapidity distributions for single diffractive (SD) and non single diffractive (NSD) events selection [45].

In the next chapter we will describe our tune using `mcnntunes` and in it we used the same distribution listed here. All the graphs for these observables are then shown with the final tunes descriptions.

<sup>1</sup>Note that the parameters related to the simulation of the hadronization and beam remnant are set to the values of the Monash tune

### 4.3.2 Pyhtia configuration and the tunes

On the top section of Table 4.2 and Table 4.3 are reported the values for the `pythia8` parameters used in the CP tunes and, on the bottom, the five parameters resulting from the tune.

PYTHIA8 parameter	CP1	CP2
	NNPDF3.1 LO	NNPDF3.1 LO
PDF Set		
$\alpha_s(m_Z)$	0.130	0.130
SpaceShower:rapidityOrder	off	off
MultipartonInteractions:EcmRef [GeV]	7000	7000
$\alpha_s^{\text{ISR}}(m_Z)$ value/order	0.1365/LO	0.130/LO
$\alpha_s^{\text{FSR}}(m_Z)$ value/order	0.1365/LO	0.130/LO
$\alpha_s^{\text{MPI}}(m_Z)$ value/order	0.130/LO	0.130/LO
$\alpha_s^{\text{ME}}(m_Z)$ 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 4.2:** CP1 and CP2 tunes settings are report here together with the values for the parameters tuned. CP1 and CP2 use a LO PDF set. CP1  $\alpha_s$  is different between matrix element calculation and MPI that use a value of 0.1365 ISR and FSR that instead use 0.130. While, CP2 use the same value for all processes, it is fixed at 0.130. In both cases  $\alpha_s$  run with a LO evolution. Table from [39]

PYTHIA8 parameter	CP3	CP4	CP5
	NNPDF3.1 NLO	NNPDF3.1 NNLO	NNPDF3.1 NNLO
PDF Set			
$\alpha_s(m_Z)$	0.118	0.118	0.118
SpaceShower:rapidityOrder	off	off	on
MultipartonInteractions:EcmRef [GeV]	7000	7000	7000
$\alpha_s^{\text{ISR}}(m_Z)$ value/order	0.118/NLO	0.118/NLO	0.118/NLO
$\alpha_s^{\text{FSR}}(m_Z)$ value/order	0.118/NLO	0.118/NLO	0.118/NLO
$\alpha_s^{\text{MPI}}(m_Z)$ value/order	0.118/NLO	0.118/NLO	0.118/NLO
$\alpha_s^{\text{ME}}(m_Z)$ 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 4.3:** here are reported CP3, CP4 and CP5 tunes settings, and the results for the tune. The three tunes use an equal  $\alpha_s$  value for al the processes,  $\alpha_s = 0.118$  running with a NLO evolution. The difference between CP3 and the other two tune is that CP3 use a NLO PDF set while CP4 and CP5 a NNLO one. CP5 ISR emission is also ordered according to rapidity. Table from [39]

Below we are going to focus on CP5 tune. We are going to reproduce this tune using the same settings for `pythia8` but using a different tuning software: `mcnntunes`.

Next chapter discusses the general approaches to the tune and why we use machine learning and and on the description of the main tool used for the tuning in this work MCNNTUNES.

# Chapter 5

## Tune procedure and MCNNTUNES

The study of the underlying event, or more in general of soft QCD processes requires the use of Monte Carlo generators that often are based on phenomenological models with lots of free parameters. These free parameters must be tuned on experimental data in order to obtain meaningful results from the simulations. The procedure of estimate the best parameters values in order to best reproduce the observables under analysis is called *tune*.

The tune 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 term of computational time required for just a single job.

To tune a certain number of parameters the number of jobs you have to run increase with this number of parameters. In fact, the parameters space dimension increase with the number of the sampling we want to perform and the granularity we want 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 require 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 computational expensive and a scan in a 5 parameters space with 10 division 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 CondorHT (CERN).
- 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 right approach that is used in the high energy physics tune and that we are going to use in the follow for our tune.

Let us discuss this last approach with more details in the next section.

## 5.1 Parametrization-based approach

The parametrization-based approach is the most used method. The current state-of-art in the tune 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 have been implemented in the software PROFESSOR [46].

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 use a polynomial fitted to well describe the real function).

$$h(p) \xrightarrow{\text{parametrization}} \bar{h}(p) , \quad (5.1)$$

where the real function have been substituted by the surrogate one.

After that, a *loss function*  $\mathcal{L}(\bar{h}(p), h_{\text{data}})$  is defined between the surrogate function and the experimental data. A common choice for it is the  $\chi^2$  function defined as:

$$\mathcal{L}(\bar{h}(p), h_{\text{data}}) \equiv \chi^2 = \frac{(\bar{h}(p) - h_{\text{data}})^2}{\sigma^2} . \quad (5.2)$$

In the end to find the best parameters estimation, this loss function need to be minimized. The set of parameters  $p_{\text{best}}$  that do this are the best evaluation that our generator can provide for the real values and we are going to call this set of best parameters: *tune*.

$$p_{\text{best}} = \arg \min_p \mathcal{L}(\bar{h}(p), h_{\text{data}}) . \quad (5.3)$$

In our study instead of the common software PROFESSOR based on the polynomial parameterization we use the machine learning approach implemented in MCNNTUNES software [47] using Feed Forward Neural Networks. MCNNTUNES is a software developed by S. Carazza, S. Aioli and M. Lazzarin presented in [48] based on machine learning library TensorFlow [49]. MCNNTUNES is written in python and it uses neural networks (NNs) that are trained in order to learn the generator behavior to the parameters variations. This remove the polynomial constraint in the fit of the generator respond function in fact one of the main feature of the Neural Network is that they are universal function approximators.

Let's make a brief introduction on machine learning and in particular on neural networks in order to understand why this choice.

## 5.2 Machine Learning and Neural Networks

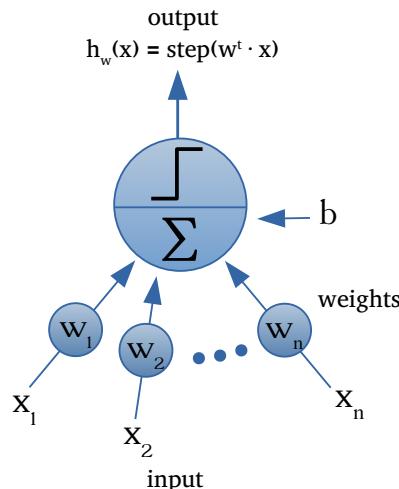
Machine learning (ML) is a particular type of Artificial Intelligence it consists in systems that learn automatically by the data that are fed into it and not by the explicit programming of the algorithm. It is clear that ML requires the training of the algorithm in order to have a predictive output related to the problem under analysis. The training is the most important step in the ML approach in fact is this step that gives to the ML the ability to return a predictive output without it the output we get is not able to give us a meaningful result.

A particular type of ML is Deep Learning that uses neural networks with more than one layer organized in a hierarchical structure to solve the problem. This is the type of ML we are interested in. But before we start with the explanation of the simple possible neural network that can be created in order to understand the basis in the logic of each unit that is going to compose the neural network.

### 5.2.1 Neural Networks - Perceptron

The concept of the Neural Network (NN) was developed in 1958 by Frank Rosenblatt. He introduce the simpler example of NN: the perceptron [50]. A representation of a perceptron is shown in Fig. 5.1 the input values are weighted and summed, an additional offset  $b$  can be introduced, then the weighted-sum is passed to an activation function (step function). So, the output that the perceptron returns is:

$$h(x) = \text{step}\left(\sum_j w^j x_j + b\right) \quad (5.4)$$



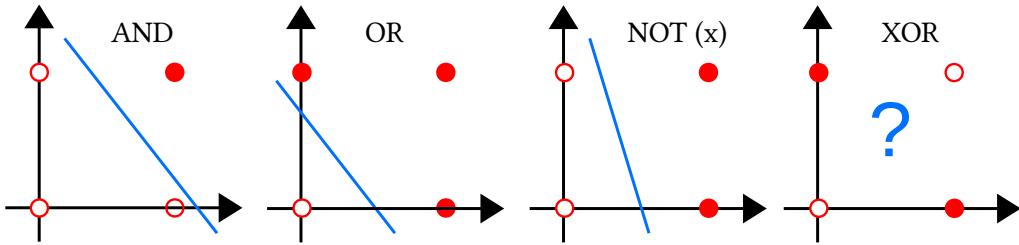
**Figure 5.1:** A schematic representation of a perceptron.

The revolutionary feature of the perceptron was the ability of learning by an adjustment of the weights. But a single perceptron is not enough this kind of logical units have lots of limitations. An example of limitation for the perceptron is shown in Fig. 5.2 where the impossibility of implement 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 acceptance of the hypothesis.

The structure is very simple with a single unit but is not enough it have a lot of limitation so we have to introduce the concept of NN with more than one unit if we want eliminate these limitations and approximate every type of functions.

### 5.2.2 Feed-Forward Neural Networks

In a Neural Network different units called "neurons" are linked together. Different type of NNs exist and they are classified according to the various ways the neurons



**Figure 5.2:** The figure shows one of the limitations of the perceptron. The XOR operation is not possible with a linear cut.

are linked each other. We are interested in *fully-connected Feed-Forward NNs*, that are the ones used in MCNNTUNES.

In fully-connected NNs each neurons from a layer are connected to every neurons in the next layer. While, the Feed Forward attribute refers to the fact that the NN have not internal recursions (loops) between neurons but all the neurons from a layer are connected forward to the ones of the next layer.

Fig. 5.3 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 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  $\vartheta_j$  as input, compute the weighted sum  $\sum_k w_{kj}x^k$  and apply the activation function  $\phi$  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:

$$f^{(i)}(x) = \sum_{j=1}^{N^{(i)}} \varphi \left( \sum_k w_{kj}^{(i)} x^k + \vartheta_j^{(i)} \right) , \quad (5.5)$$

where  $N^{(i)}$  is the number of hidden units in the ( $i$ )-th layer.

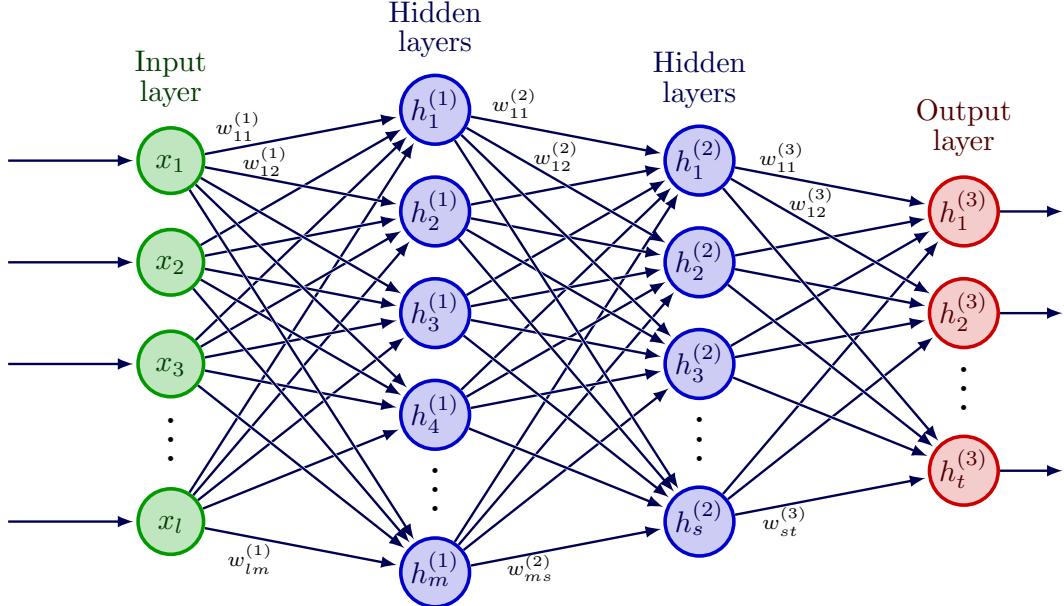
One of the biggest feature of the NNs is that they are universal function approximators [51, 52] the only request for this is that a sufficient number of hidden layers is available. This is the main reason why we want use a NN based approach instead of the polynomial one

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 do this and so get some predictive results from the NN a learning algorithm have 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 change the weights,  $w_{jk}$ , and the offsets,  $\vartheta_j$ , in order to minimize a loss function usually defined as the mean squared error ( $E$ ):

$$E = \frac{1}{2} \sum_i (h_i(x^j, w_{jk}) - d_i)^2 , \quad (5.6)$$

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.



**Figure 5.3:** 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 through 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).

In the back-propagation algorithm the weight and the coefficients are update using the *steepest-descent minimization*:

$$w_{jk}^{(i+1)} = w_{jk}^{(i)} - \lambda \left( \frac{\partial E}{\partial w_{jk}} \right)^{(i)} ; \quad \vartheta_j^{(i+1)} = \vartheta_j^{(i)} - \lambda \left( \frac{\partial E}{\partial \vartheta_j} \right)^{(i)} \quad (5.7)$$

where  $\lambda$  is the learning rate and is a user-tunable free parameter and it controls how much change the weights each time the weights are update. It is one of the main parameter in the neural networks a to small value for it can lead to a failure in the training procedure while a to large one can lead to unstable results.

The mini-batch are subset of our 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 parameters: smaller batches are faster to compute but the gradient direction is not the real one just an approximation; while bigger batch size give a good approximation of the direction of the steepest-descent but can be computational expensive.

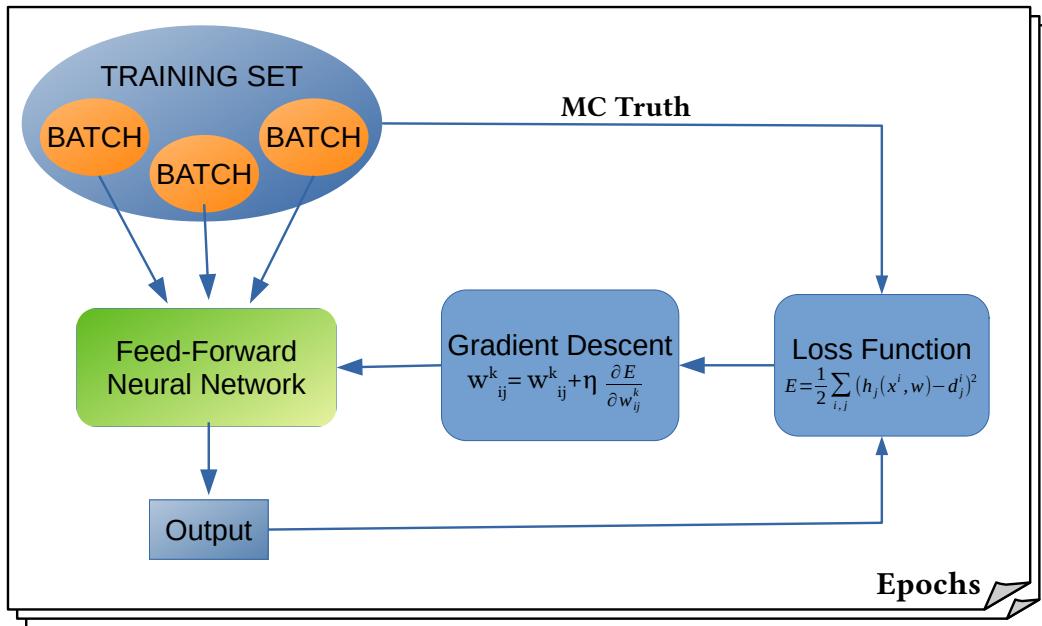
The number of evaluation of the entire training set is called epochs and it is tunable by the user.

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 we use a batch size of 25 it requires only 2000 iterations to run over all the training set.

A schematic representation of the training process is displayed in Fig. 5.4 where the training set is subdivided in 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 as *hyperparameters*.



**Figure 5.4:** The training set used to train the NN is divided in mini-batch (the mini-batch size is a user tunable parameter) than one by one the batches are feed 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. 5.7 with the back-propagation algorithm. This is done with each mini-batch in the training set and for a number of epochs defined from the user.

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

### 5.3 MCNNTUNES

`mcnntunes` [48] is a Shower Monte Carlo generators tuning tool that implements a tune procedure based on the use of Feed Forward Neural Networks (FFNNs). The advantage of using FFNNs has been described above and it is that they are universal function approximators at the simple cost of have 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 approach similar to the one in `professor` but

where the response of the generator is parameterized using these FFNNs, one for each bin; the later 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 try to infer the parameters value starting from the experimental values of the bins in the distributions used.

### 5.3.1 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 we use the parameters space shown in Table 4.1), than the generator is run with every sampled configuration. All these MC runs are going to build our dataset that is called *training set*.

A schematic explanation of how the sampling phase and the training set generation were performed in CMSSW environment (lxplus CERN) is shown in Fig. 5.5. In more detail the sampling starts by means of MCNNNTUNES script called MCNNTEMPLATE with a sampling of the defined parameter space. The sampling generates  $N$  different configurations for the parameters to tune, which 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 runcards are saved the PYTHIA generator is run with every configuration generated and this is performed not locally but on a computer's batch (e.g. the CERN one CondorHT) in order to get all the results in a few days. The runcards we pass to Condor contain also the information on the analysis we want 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, which is a standard type of data file for these analyses. The set of these YODA files, containing the information on the MC runs, composes our training set that can then be used for the tuning procedure.

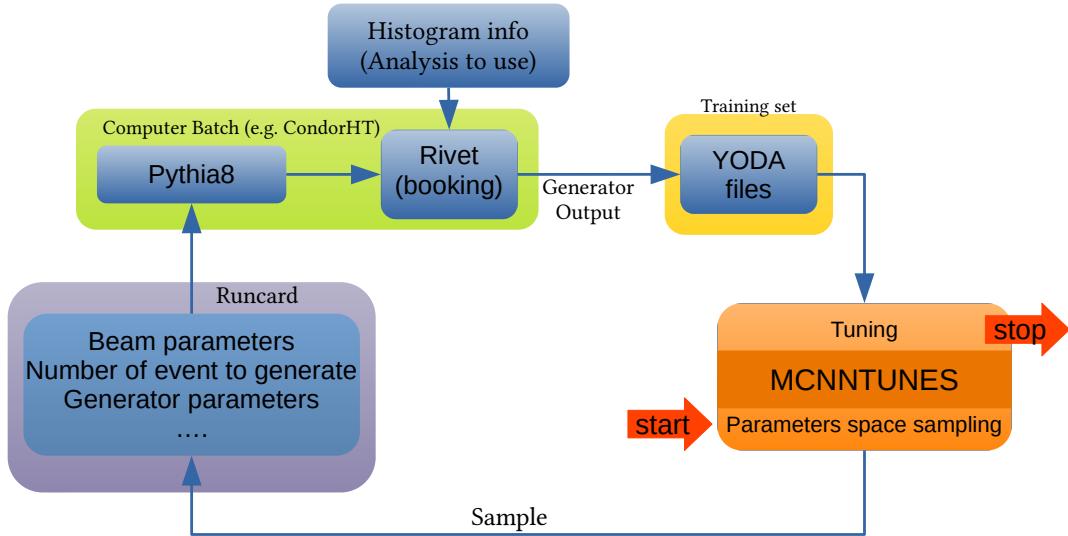
MCNNNTUNES offer also the possibility of changing the value of the hyperparameters. The values that can be modified for the Neural Network. It is possible to choose the NN architecture: the number of hidden layers, the number of neurons for each layer and the activation function used. It is also possible to set the number of epochs, the batch size and the learning rate in order to have the best training 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 we are going to discuss the procedure used for the Inverse Model in order to search for the best hyperparameters configuration.

### 5.3.2 Per Bin Model

PerBin Model is a parametrisation-based method. The main idea, as shown in Fig. 5.6 is to build a model (i.e. a neural network) for each bin in order to parameterize the generator output. Each NN takes the parameters 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



**Figure 5.5:** 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.

parameters estimation. This step define a surrogate loss function for the tuning problem. In fact, the parameterization step return a model  $h^{(i)}(\mathbf{p})$  for each bin,  $i$ , where  $\mathbf{p}$  is the vector of the parameters.

Then, this surrogate loss function defined as:

$$\chi^2 = \sum_{i=1}^N \frac{\left(h^{(i)}(\mathbf{p}) - h_{exp}^{(i)}\right)^2}{\sigma_{(i)}^2} \quad (5.8)$$

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 minimize this  $\chi^2$ .

### Errors evaluation

MCNNTUNES PerBin model as introduced in [48] did not have a proper error evaluation. 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. Thanks to S. Carazza and with the help of M. Lazzarin we had the opportunity of handle the code and add a proper errors estimation.

Now, the errors evaluation for the PerBin Model is given by the definition of a confidence interval using the  $\chi^2$  function. In fact, as shown in section 9.6 and 9.7 of [54], for an estimators vector  $\hat{h}(\mathbf{p}) = (\hat{h}^{(1)}(\mathbf{p}), \hat{h}^{(2)}(\mathbf{p}), \dots, \hat{h}^{(n)}(\mathbf{p}))$  for the parameters  $\mathbf{p}$  the probability distribution function and the likelihood (the  $\chi^2$  in our case) in limit

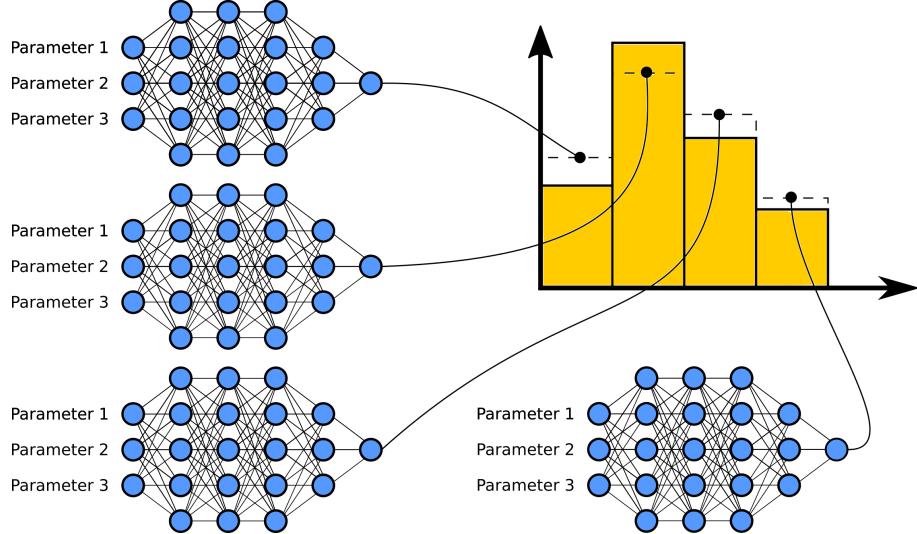


Figure 5.6: Figure from [48]

percentile	$Q_\gamma$				
	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
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 5.1:** The table report the values of the quantile  $Q_\gamma$  for different confidence level 0.683 is the row corresponding to the  $1\sigma$  definition and is the one of our interest.

of a large sample<sup>1</sup> are Gaussian distributed. The probability distribution function for the estimators is than:

$$f(\hat{h}(\mathbf{p})|h(\mathbf{p})) = \frac{1}{(2\pi)^{n/2}|V|^{1/2}} \exp \left[ -\frac{1}{2} (\hat{h}(\mathbf{p}) - h(\mathbf{p}))^T V^{-1} (\hat{h}(\mathbf{p}) - h(\mathbf{p})) \right], \quad (5.9)$$

where  $T$  is the transpose vector and  $V^{-1}$  is the inverse covariance matrix. Can be shown that also the likelihood is Gaussian as the probability distribution function. So, we can define our confidence interval using the  $\chi^2$  statistic as:

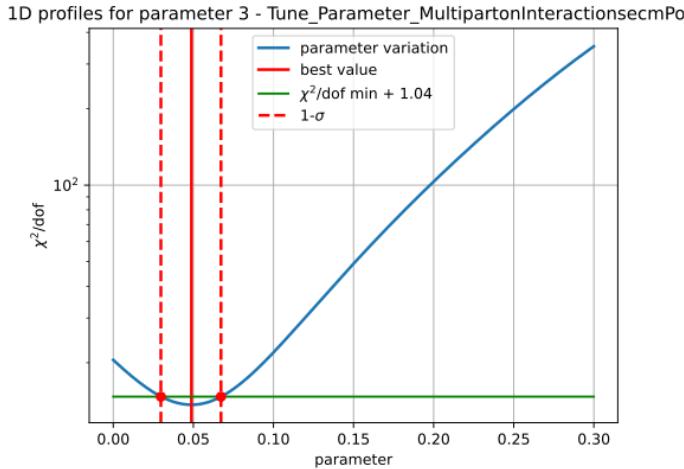
$$\frac{\chi^2(\text{c.i.})}{N_{dof}} = \frac{\chi^2_{min}}{N_{dof}} + \frac{Q_\gamma}{N_{dof}} \quad (5.10)$$

The variation is dependent on the number of parameters and on the chosen confidence level ( $1\sigma = 0.683$  in our case) and a list of the values is reported in Table 5.1. Then, the error is defined as the value of the parameters that give a deviation from the minimum value of the  $\chi^2/N_{dof}$  equal to the  $Q_\gamma/N_{dof}$  value for a confidence level of 0.683, as defined in Eq. 5.10.

An example for the evaluation in `mcnnntunes` is shown in Fig. 5.7 where the green line is the deviation from the minimum value of the  $\chi^2$  defined in Eq. 5.10 and the

<sup>1</sup>In other case this still a good approximation.

errors are given by the points where the  $\chi^2/DoF$  reach these value. Therefore, the intersections between green and blue line that is the  $\chi^2/N_{dof}$  for the different values of the parameter.



**Figure 5.7:** This figure shows the error evaluation in `mcnntunes` for the `MultipartonInteractions:ecmPow Pythia8` parameter. The blue line is the value of the  $\chi^2/N_{dof}$  evaluated for different parameter values. The best estimation is indicated by the vertical red line, while the green line is the quantity in Eq. 5.10. The error is evaluated from the intersection of blue and green lines.

### 5.3.3 Negative aspects

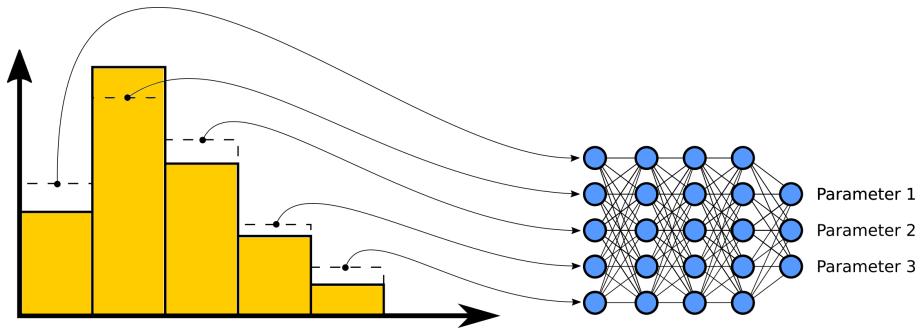
One of the negative aspect is that this model is more computational expensive than the below discussed Inverse model. This is due to the large number of NNs built and trained from this model. The high cost in term of time required to get the model work do not give the possibility for a scan in the hyperparameters space in order to obtain the best configuration for the NN architecture.

This can impose some limitation on the model performance that cannot get its maximum performance.

### 5.3.4 Inverse Model

The Inverse Model is the most innovative tuning procedure introduced by `mcnntunes`. This model contrarily to the PerBin Model takes the histograms bins as input and returns parameters values as output. For the Inverse Model the NN used is only one as shown schematically in Fig. 5.8. What the Inverse Model try to do is to learn the inverted model of the generator. So starting from the observed values the model try to reproduce the parameters values necessary to get the histograms we use as input.

The model is build 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.



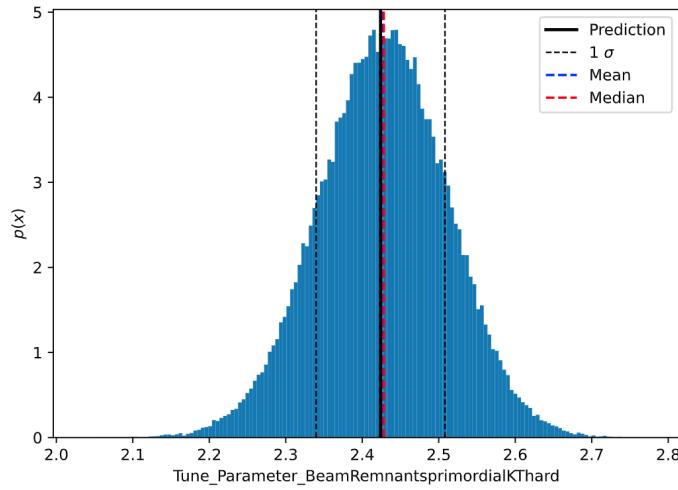
**Figure 5.8:** Figure from [48]

### Errors evaluation

The errors are evaluated in a different way respect to PerBin Model in fact in the Inverse Model there is not 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. 5.11 with a diagonal covariance matrix that have experimental uncertainties on the main diagonal.

$$f(x_i; h_{\text{exp}}^{(i)}, \sigma_{\text{exp}}^{(i)}) = \mathcal{N} \cdot \exp \left[ -\frac{1}{2} \sum_{j=1}^{N_{\text{bins}}} \frac{(x_i - h_{\text{exp}}^{(i)})^2}{\sigma_{\text{exp}}^{(i)}{}^2} \right] \quad (5.11)$$

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. 5.9, from this distribution one can compute the error by evaluating the standard deviation.



**Figure 5.9:** Predictions spread for the inverse model after that a Gaussian resample is performed

Note that this is a new method for the tune. As we are going to see this method requires more attentions than the PerBin Model to get it working correctly in the case of an high number of parameters to tune.

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 the *hyperparameter optimization* it is required to get the method working.

The procedure consist in build 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. Than 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 computing a loss function between the predicted value and the Monte Carlo truth defined as:

$$L = \sum_i \frac{|p_i^{\text{true}} - p_i^{\text{pred}}|}{p_i^{\text{true}}} \quad (5.12)$$

The configuration that minimize the distance between predicted and real values (MC truth), and so this loss function, is the best architecture in the selected hyperparameters space.

Once the best model is found, it is retrained using the MC runs contained in both training and validation set. 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. 5.10.

The hyperparameters scan is performed using the python package `hyperopt`.

## Problems

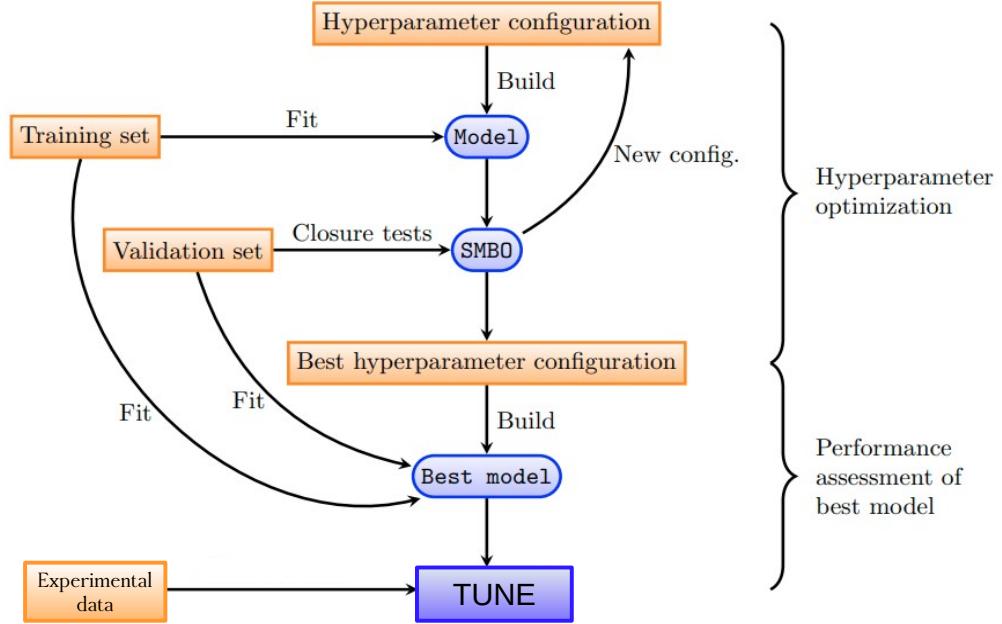
As we are going to see the main problem for the Inverse model is the stability in the results. The operation that this model is trying to do is very hard, learn the inverse response of the generator is not an easy task. These difficulties in some case lead to a bad prediction.

As we are going to see in the chapter that describe our tune the Inverse model requires some more attention than the PerBin model to get it work properly and in some case when the number of parameters became larger and maybe the correlation between the parameters are important this model fails.

But if more care are given to this model this can become a very powerful method for future tunes.

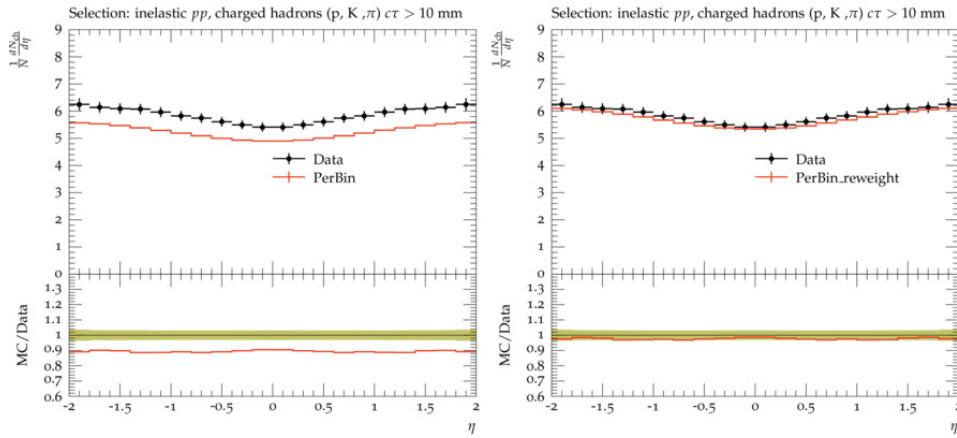
### 5.3.5 Weightrules in MCNNTUNES

MCNNTUNES as PROFESSOR implement the possibility of change the weight of the singles bins in the distributions. This is a really useful feature. As we will discuss later we use these weightrules to get a results more similar to CP5 in our tune. This feature gives to us the possibility to decide which distribution is more important in our tune. Increasing the weight of a bin this bin became more important in the overall  $\chi^2$  evaluation and it is better described by the simulations. An example is



**Figure 5.10:** The hyperparameter optimization procedure is schematically shown here. The model is trained using a training set than 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 the network and the best parameters are estimated. Figure from [48]

shown in Fig. 5.11 where the application of the weightrules takes us to a better description of the experimental data.



**Figure 5.11:** An example of weightrules application. Here the black point are the experimental data from the CMS analysis at  $\sqrt{s} = 13$  TeV [42] while the colored lines are the simulations, the vertical lines on MC points indicate the statistical uncertainties. It is easy to see that the distribution in the right panel is not well described from the tune, but thanks to the weightrules we can give to this distribution a greater importance in the tune and describes it better as shown in the left panel. This is better explained in the next chapter.

# Chapter 6

## Our Tune for the Underlying Event

This section focus on our work in order to reproduce a similar tune to CP5 one for the underlying event and minimum bias obseravbles. This is done to test the ability of MCNNTUNES of being one valid tool for the tuning of Monte Carlo generator with real data.

In order to validate MCNNTUNES as a good tool for the tuning we decide to firstly performer a simpler tune with only two free parameters and then try to reproduce CP5 [39] with all the five parameters variation.

### 6.1 Introduction

We have performed a different tune for the underlying event and minimum bias observables using the same distributions, listed in Chapter 4.3.1 and used in CP5 tune, so we expect to get a similar result from the tune using MCNNTUNES.

Just a quick reminder for the PYTHIA8 settings used in CP5 and in our tune:

- The PDF set used is NNPDF3.1 calculated to the NNLO [55];
- an  $\alpha_s$  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 we employed both the MCNNTUNES operation modes described in the previous chapter: PerBin and Inverse models.

### 6.2 First test: only two parameters variation

The first simplified test we perform is the tuning varying of only two parameters. The parameter chosen are the `MultipartonInteractions:pT0Ref` and `MultipartonInteractions:ecmPow`. These two parameters have been introduced in Section 3.1. The aim of this simplified test is to check the correct operation of MCNNTUNES in fact in a restricted parameters space it is simpler to learn the generator behavior and predict the corrects best values for the parameters.

Table 6.1 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).

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

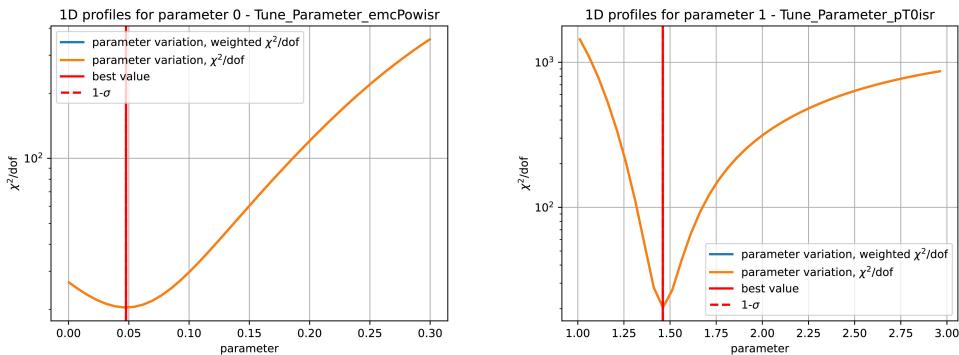
**Table 6.1:** Parameters space for the two parameters test. The other parameters are set to the CP5 default values.

Note: during this first part we don't use the two distributions related to the single diffractive and non single diffractive event selection for a bug on the routine of the analyses that have been fixed up before the real tune.

Let now discuss the results obtained for the test with the two models.

### 6.2.1 Per Bin Model results

The result we obtain for the PerBin model are reported in Table 6.2. The PerBin model estimation of the best parameters is performed by a loss function minimization. The output of the minimizer is reported in Fig. 6.1, the loss function ( $\chi^2/\text{dof}$ ) is the orange one, while the best parameter is marked by a solid red line.



**Figure 6.1:** 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  $\chi^2/\text{DoF}$  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 6.2 with a comparison on CP5 boundary for each parameter. To be notice that the errors on the parameter estimated with PerBin model are not reported here because a correct error implementation, as the one described in Section 5.3.2, was not yet implemented in the software at the moment of this test, but here we can see an example of the old error estimation. The old errors estimation process was not working in fact the errors lines (dashed red lines) are overlapped to 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.

But as we expected the results we get are similar to the ones obtained from the

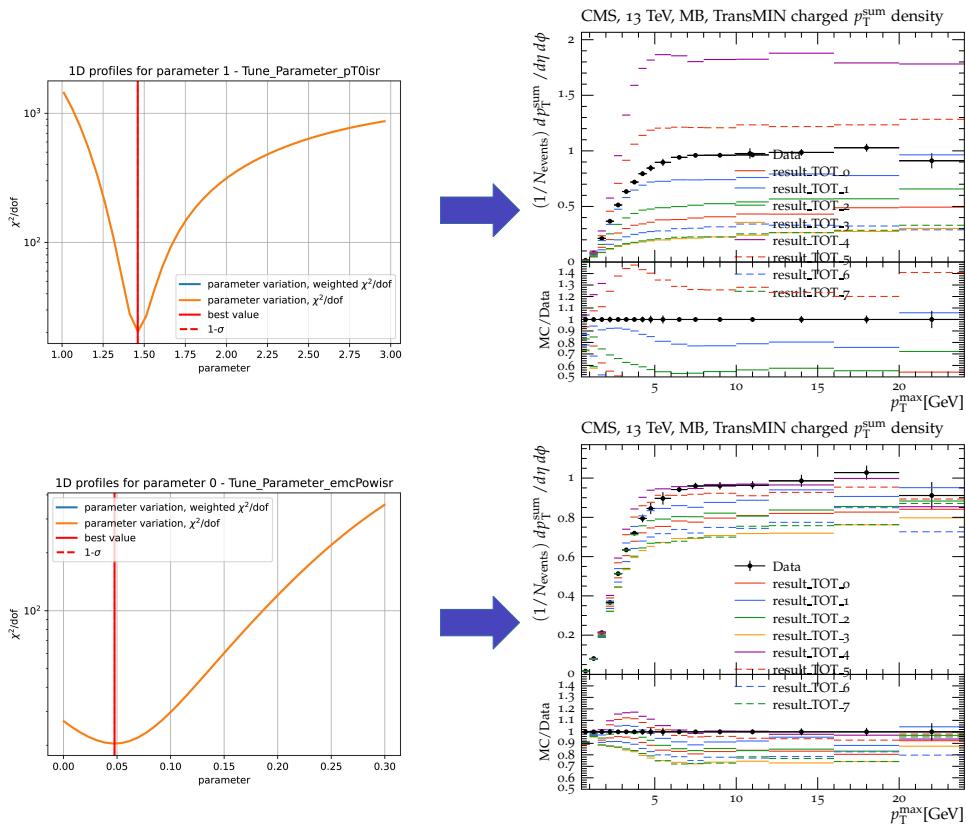
CP5 tune. The distributions obtained from the simulation using this parameters are shown below (Section 6.2.3) together with results from Inverse model and CP5.

Parameter	Value	CP5 (down & up)
MultipartonInteractions:pT0Ref	1.46064	1.41 – 1.46
MultipartonInteractions:ecmPow	0.04771	0.03

**Table 6.2:** Results for the PerBin model in two parameter variation test. The error evaluation was not correctly implemented yet and so the PerBin errors omitted in this table. The upper and lower limit for CP5 are also reported here for a direct comparison between the two tunes.

An important observation is that our model is more sensible to some parameter respect to others.

As an example, in Fig. 6.2 it is clear that our model is more sensible to the parameter `MultipartonInteractions:pT0Ref` (top) respect to `MultipartonInteractions:ecmPow` (bottom). This is reflected in the output of the minimizer a greater sensibility gives a better defined minimum and vice versa. The top two panels show what happen when the distribution are very sensitive to the variation of the parameter as we can see on the left 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. 6.2 where the distribution is less sensitive to the variation of the parameter `ecmPow` and so the minimum is less defined and so we are going to have a larger error on the estimation.



**Figure 6.2:** the sensibility of the tuned distributions respect to the variation of the parameters in 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 point are the data.

### 6.2.2 Inverse Model results

The other operation mode offered by MCNNTUNES is the Inverse model. The results we get from the first test of the Inverse model are described here while the overall distribution are reported in next section together the ones from the PerBin Model. As mentioned before to make this model work properly one have to perform a hyperparameters optimization. our hyperparameter optimization was a scan of the architecture parameters shown in Table 6.3 the number of trials (combinations of these parameters) was 1000. in this case the best model we found is the one reported in Table 6.4.

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

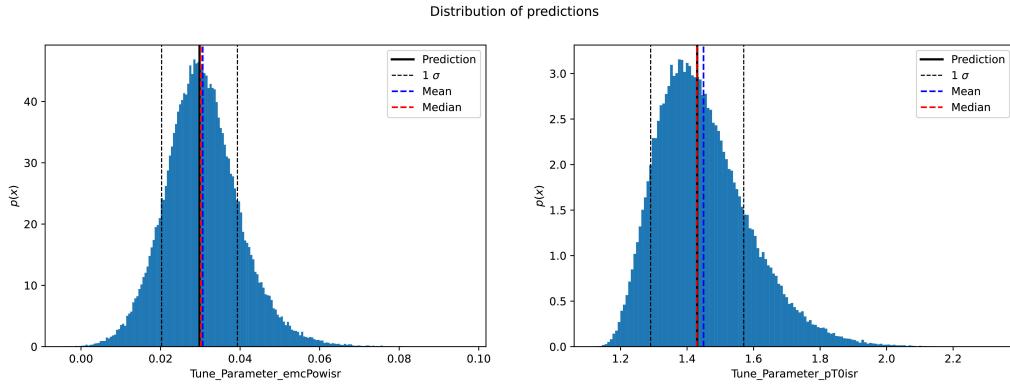
**Table 6.3:** Hyperparameter space scanned for the optimization of the NN architecture.

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

**Table 6.4:** Best hyperparameters model found for the test with 2 parameters variation.

Once the best model is trained the output we get 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. 6.3 these are obtained from the re-sampling phase using the multivariate Gaussian distribution described in Eq. 5.11.

The estimated parameter are marked by the solid black line, while the dotted black lines are the error on the parameter. The value we get from the tune are also reported in Table 6.5 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 6.3:** The spread of predictions we get 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.

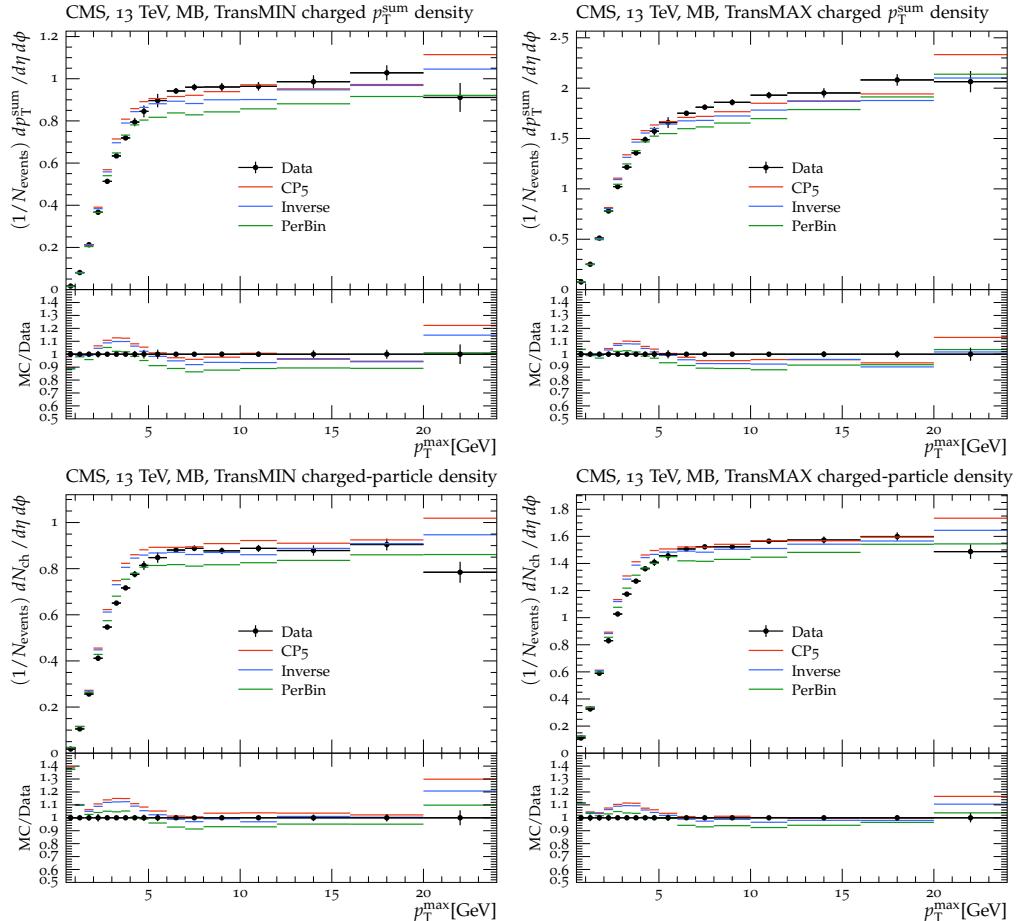
Parameter	Value	CP5 (down & up)
MultipartonInteractions:pT0Ref	$1.43 \pm 0.14$	$1.41 - 1.46$
MultipartonInteractions:ecmPow	$0.0298 \pm 0.0095$	0.03

**Table 6.5:** Results for the Inverse model in two parameter variation test. The upper and lower limit for CP5 are also reported here for a direct comparison between the two tunes. The two predicted parameters are compatible to the one in CP5.

### 6.2.3 Overall results

In this section we are going to show some overall results for the first test with only two free parameters. We are not going to show all the graphs here we list them all in the appendix Appendix A.

From the graphs in Fig. 6.4 is clear that MCNNNTUNES gives some good result for the test.



**Figure 6.4:** Results for the two parameters variation test. Here are reported the distributions from the  $\sqrt{s} = 13$  TeV CMS analysis [34] that show the trans-MAX 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 data are compared to the MC prediction from the result we get using PerBin Model (green line), Inverse Model (blue line) and the existing tune CP5 (red line). The data are well described by all the tunes. Our tune describe very well the low- $p_T$  region ( $p_T \lesssim 5$  GeV).

The result we get from PerBin model (green line) and from Inverse model (blue line) are similar to the result obtained from CP5 tune (red line). our tune describe the low region ( $p_T \lesssim 5$  GeV) very well this region is very important because is the region with lower error on the experimental data.

Overall, we can consider this test successfully passed from MCNNNTUNES, it gives

result similar to the standard tool PROFESSOR. This was only a simplified test to check the correct operation for the tool. So, passed the test we decide to extend our analysis to a complete tune for the underlying event.

## 6.3 Our tune for the UE in Minimum Bias observations

Given the good results for the test we extend our analysis to the variation of five parameters. The interested parameter 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 6.6.

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]

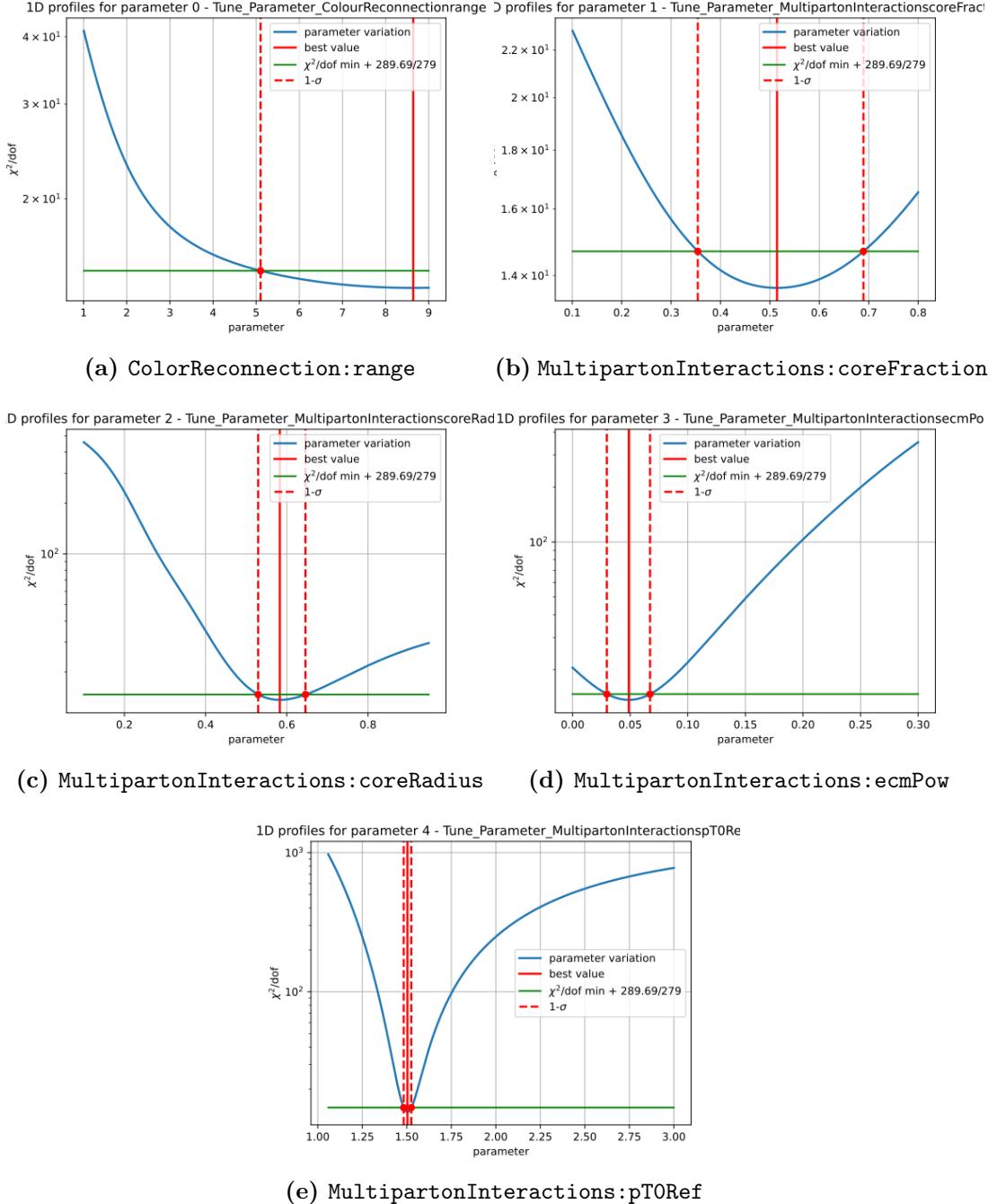
**Table 6.6:** The variation ranges for the five parameters that we want to tune. These are the same used in the CP5 tune in [39].

As the parameters space increase we need to increase also the number of samples and then the size of the training set in order to have a sufficient granularity in the sampling of the space. The training set we use for the PerBin model with five parameters variation is composed from approximately 2000 MC runs, instead for the Inverse model we try also larger training set.

### 6.3.1 Per Bin Model results

The PerBin model is the model that give to us the best results, the parameters estimation we get from the PerBin model loss function minimization is reported in Fig. 6.5. In the figure the five parameters  $\chi^2/\text{DoF}$  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 sensible parameter is the `MultipartonInteractions:pT0Ref` (6.5e) the minimum in that case is very well defined and the error small. The parameters in Fig. 6.5c and Fig. 6.5d are also well defined the error is not to big. The not so good defined parameters are the `MultipartonInteractions:coreFraction`, that is in a minimum but with a larger error than the other parameters which is why the distribution 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 indicated a saturation of this parameter, over a certain value all the possible reconnection have already occurred, and so a consequent small sensitivity to this parameter variation.

### 6.3. OUR TUNE FOR THE UE IN MINIMUM BIAS OBSERVATIONS



**Figure 6.5:** The minimizer output for every tuned parameter. The blue line is the  $\chi^2/\text{DoF}$  as a function of the parameter value. The solid vertical line indicate 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 a saturation in the parameter is reached after a certain value this lead to a small sensitivity to the parameter variation and so a non-well defined minimum. The upper limit in this case cannot be calculated. The other parameters are all in a real minimum and with a concrete error evaluation.

The value we get for the parameter are reported also in Table 6.7 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`.

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

**Table 6.7:** The results of the tune using PerBin Model. All the values are compatible with the one with the CP5 using a  $Z$  test with a significance level of 0.05. But the value predicted for the Color Reconnection have 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 exceed the scanned space for the parameters defined for each parameter in Table 6.6.

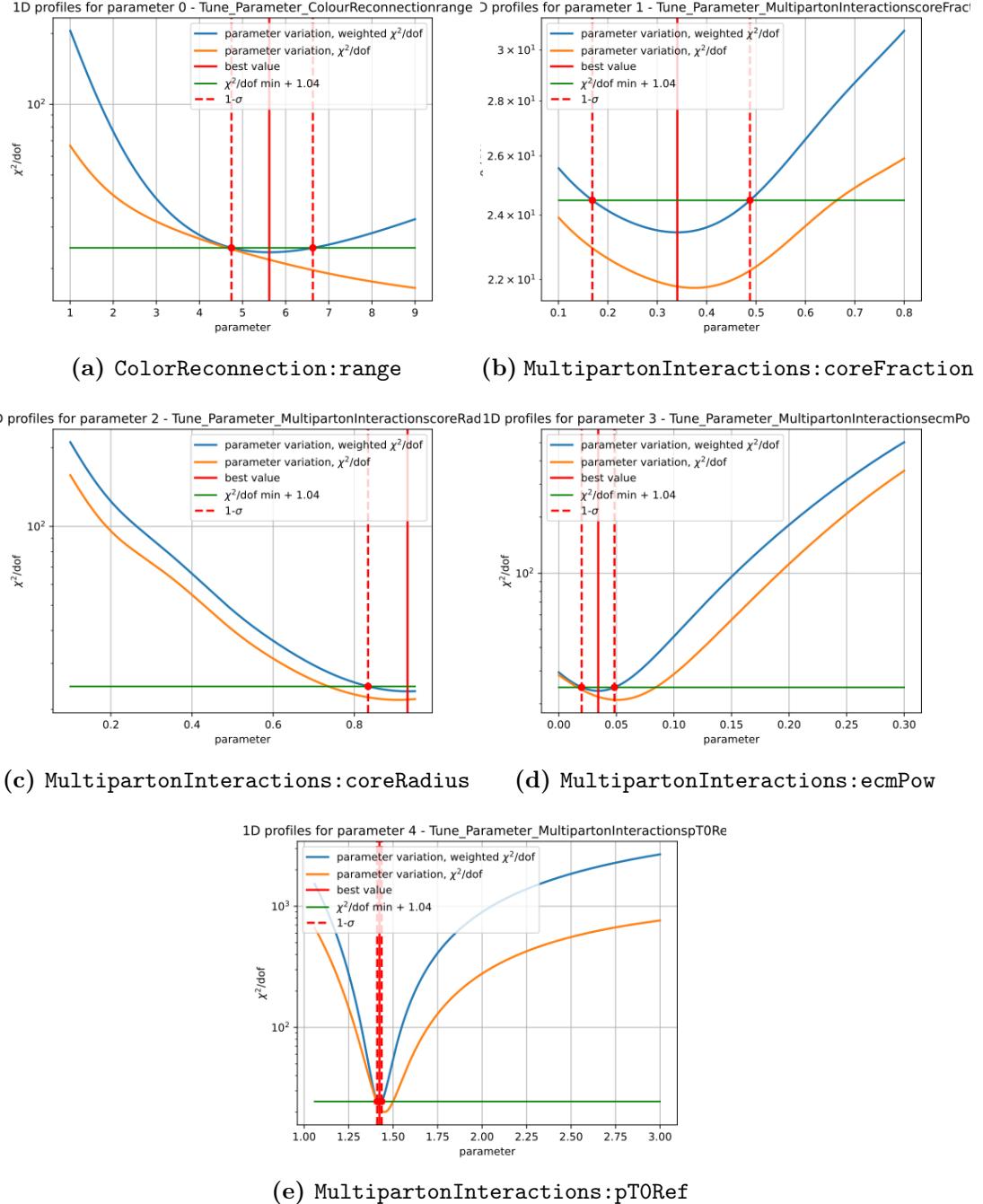
The distribution we get are reported in the Section 6.3.3 as we can see in the Fig. 6.13 the PerBin model does not describe very well this distribution for the pseudorapidity of the inelastic production of hadrons. A possible explanation is that the experimental uncertainties on the bins of this distribution are higher than the ones in others and so this distribution be less important in the overall loss function.

So, we decide to perform a second tune using PerBin Model. In the second tune we give to all Fig. 6.13 bins an higher weight using the weightrules implemented in MCNNTUNES, we give a weight of 5 to all bins of this distribution. In this way we are giving a greater importance to this distribution and so we are going to describe those data better.

The output of the minimization step is reported in Fig. 6.6. The value we get from the weighted tune are also reported in Table 6.8. The values we get are compatible also in this case with CP5 except for the `MultipartonInteractions:coreRadius` that is higher than the one predict from CP5.

But if we look at the overall result in Section 6.3.3 the PerBin Model with different weights give a result more similar to CP5 in almost all the distributions.

### 6.3. OUR TUNE FOR THE UE IN MINIMUM BIAS OBSERVATIONS



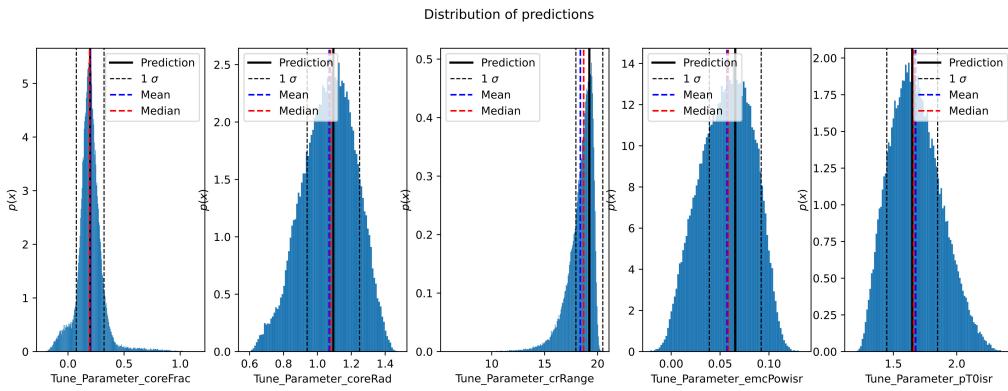
**Figure 6.6:** The minimizer output for every tuned parameter using PerBin model with the re-weight. The blue line is the  $\chi^2/Dof$  as a function of the parameter value before the re-weight while the orange line indicate the same function but applying the re-weight. The solid vertical line indicate 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.

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	$0.58^{+0.06}_{-0.05}$	$0.9^{+null}_{-0.1}$	$0.67 - 0.69$
CR:range	$8.6^{-3.5}_{+null}$	$5.6^{+1.0}_{-0.9}$	$4.88 - 4.69$

### 6.3.2 Inverse Model results

On the other hand if the PerBin model gives to us very good result, we cannot tell the same for the Inverse Model. In this case the Inverse Model gives very bad results. Also after the hyperparameters optimization, using the scan space described in Table 6.3, the model fails.

The distributions of predictions are reported for each parameter in Fig. 6.7 and the actual parameters in Table 6.9 as we 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 parameter are determined with a very large distribution and so very large errors.



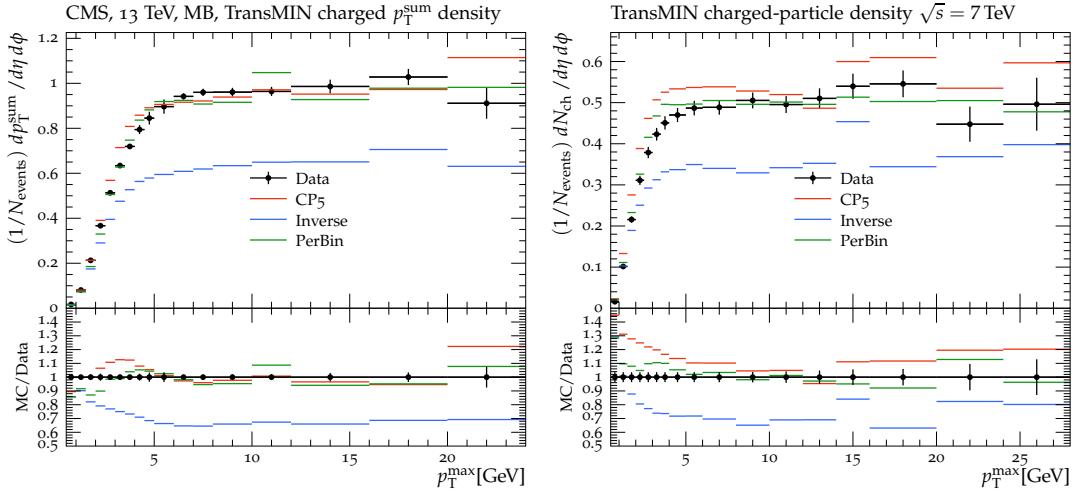
**Figure 6.7:** The spread of prediction obtained from the Inverse model. It is clear that also after the hyperparameters optimization the inverse model is not working properly. The distribution have long tails out of the limits for the variation, the central one refered to the Color Reconnection is predicted completely out of the boundaries.

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

**Table 6.9:** The predicted values using Inverse model are showed here, the model is not working in this case the predicted values for the core radius and for the color reconnection are out of the boundary while other parameter are predicted with a quite large error.

We can see from Fig. 6.8 the tune we get from the Inverse Model (blue line), as was expected, cannot describe the observables distributions. The tune misses all the experimental data points approximately by a 50% of the value. Se we have to exclude this model 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.

In MCNNTUNES presentation paper [48] is reported 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, we decide to introduce a Gaussian bias in the sampling phase instead of a uniformed distributed sampling. Our Gaussian sampling was peaked on the parameters predicted by the PerBin Model, but also this procedure does not leads to some consistent results.



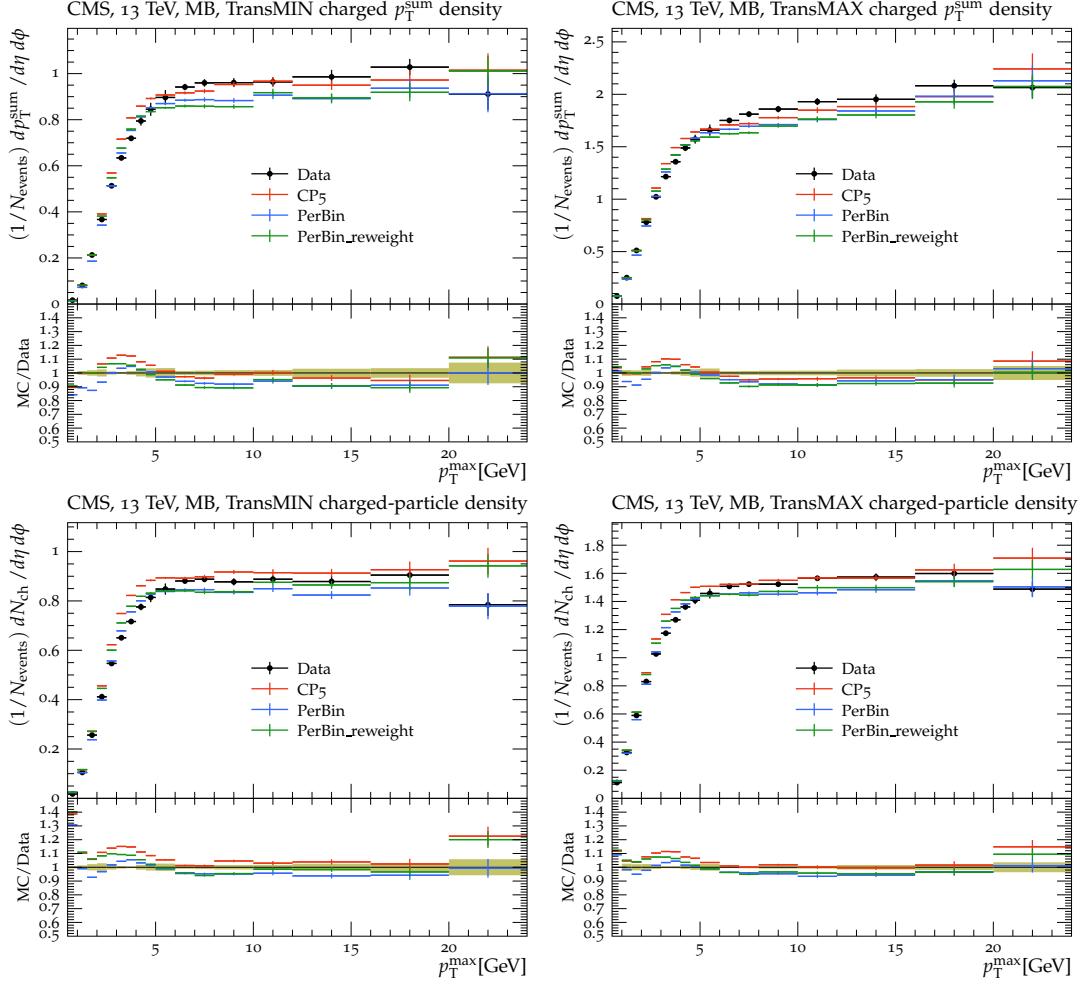
**Figure 6.8:** An example of the fact that the Inverse model (blue line) is not working: the bins are filled to only the 50% of the expected value in almost all the bins. On the left is shown the transMIN charged particle transverse momentum sum density for the CMS analysis at  $\sqrt{s} = 13$  TeV [34] and on the right the transMIN charged particle density from the CMS analysis [44] in both case as a function of the leading object transverse momentum.

### 6.3.3 Overall results

Overall, what we have are two tunes performed with the PerBin Model that are quite good tunes. 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  $\chi^2$  evaluation.

The first distributions showed in Fig. 6.9 are referred to the charged particle multiplicity and the charged particle scalar  $p_T$  sum in the two transverse regions (Trans-MAX 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 we have gotten 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 our tunes 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. 6.10 are reported the same distribution but at  $\sqrt{s} = 7$  TeV also in



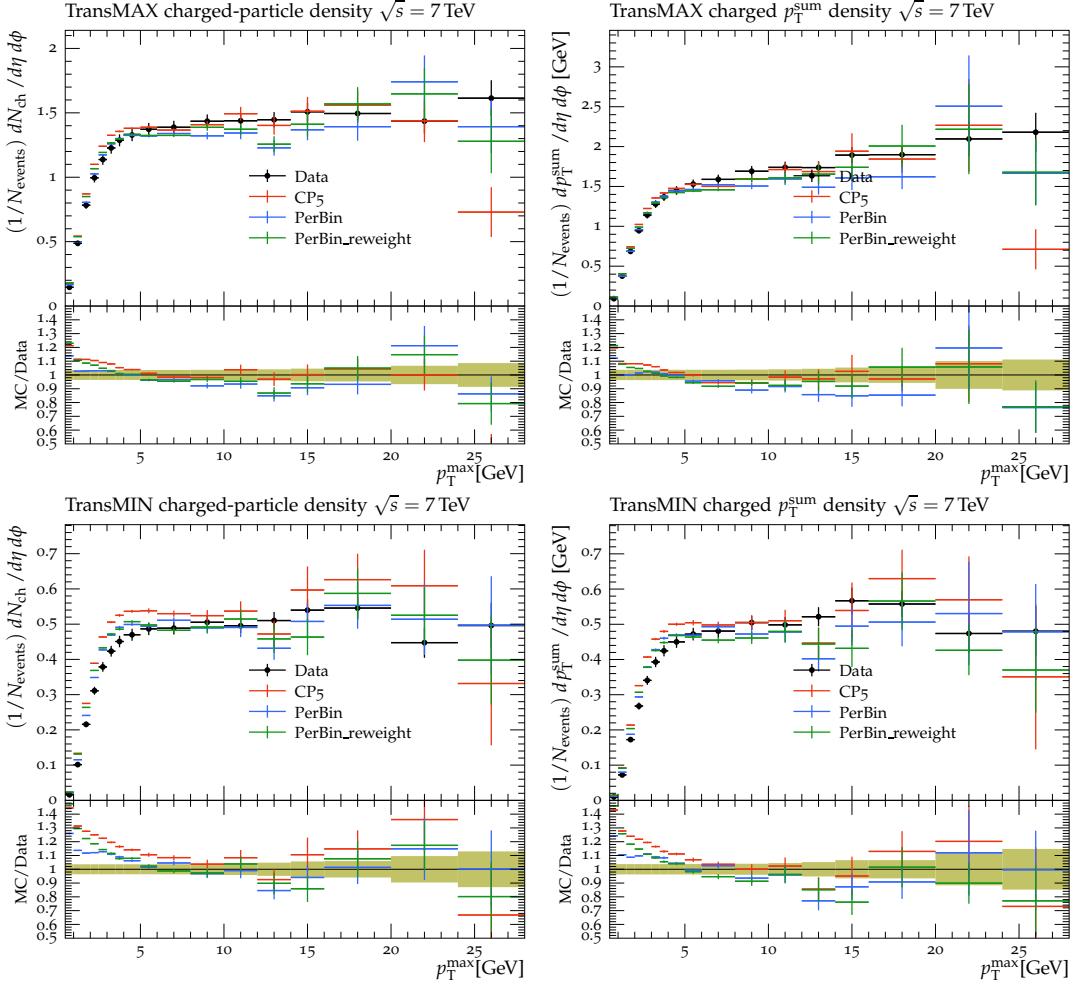
**Figure 6.9:** In this figure the data from the  $\sqrt{s} = 13$  TeV CMS analysis [34] that show the transMAX charged particle density (upper left) and the charged  $p_{\text{T}}$ -sum density (upper right); the transMIN charged particle density (lower left) and the charged  $p_{\text{T}}$ -sum density (lower right) as a function of the transverse momentum of the leading charged particle. The CP5 tune is compared to our tune using the PerBin Model. Our 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  $p_{\text{T}}$  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.

this case our tunes are good in describing the distributions, they seem 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. 6.12 and Fig. 6.13 represent the pseudorapidity distribution for diffractive events and for inelastic charged hadrons production. The left distribution of Fig. 6.12 and the distribution in Fig. 6.13 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

### 6.3. OUR TUNE FOR THE UE IN MINIMUM BIAS OBSERVATIONS



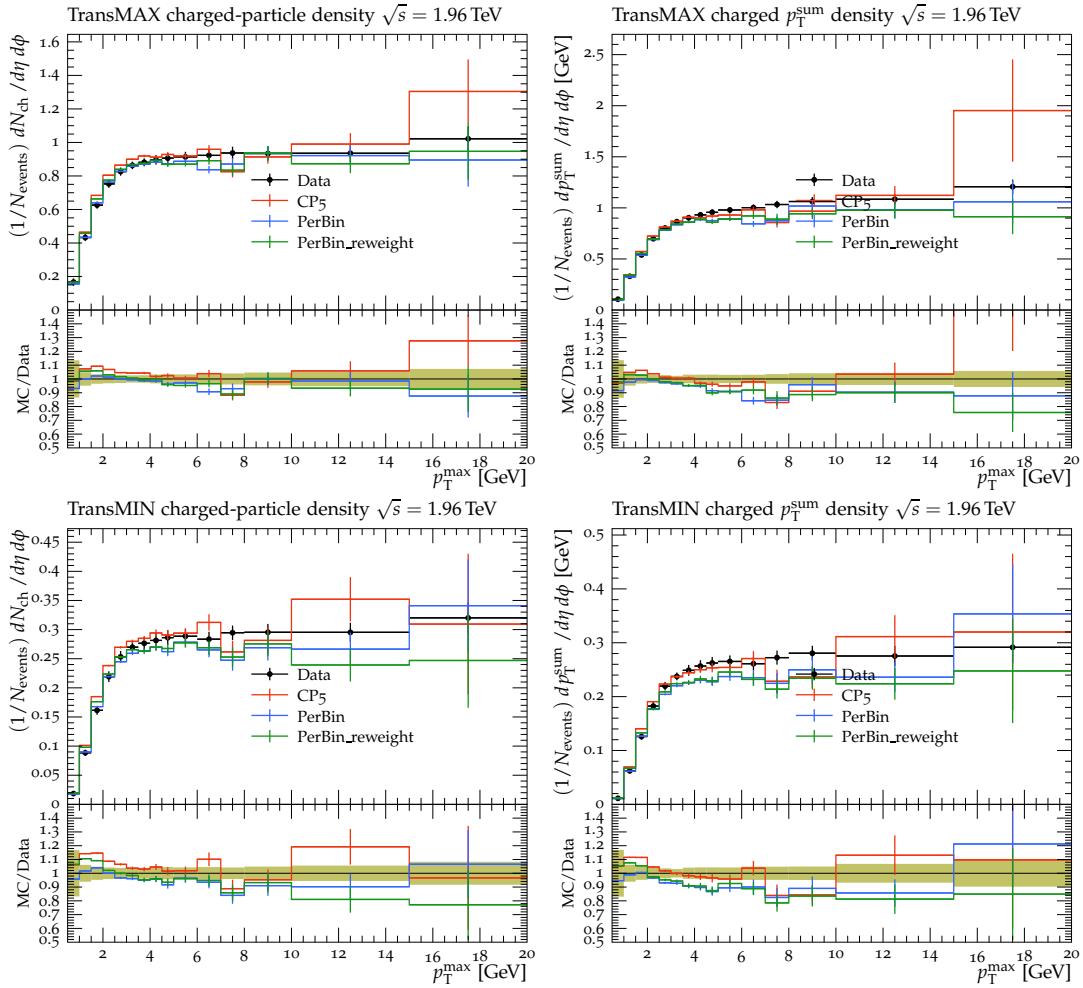
**Figure 6.10:** Here the data at  $\sqrt{s} = 7$  TeV from the CMS analysis [44] for the transMAX charged particle density (upper left) and the charged  $p_T$ -sum (upper left); the transMIN charged particle density (lower left) and the charged  $p - T$ -sum are displayed as a function of the leading object transverse momentum. The three tunes describe the data (black point) very well. The low  $p - T$  regions are described better from our tune respect to the CP5 tune. As was expected the result 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.

also in this case we have a good result from MCNTUNES tool.

A numerical evaluation on the overall difference between MC and experimental points can be evaluated for all the three tunes using the  $\chi^2/\text{DoF}$  definition:

$$\chi^2 = \sum_i \frac{(MC_i - exp_i)^2}{\sigma_i^2} . \quad (6.1)$$

The results we get are reported in Table 6.10. From this is clear that from this 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. 6.12 and



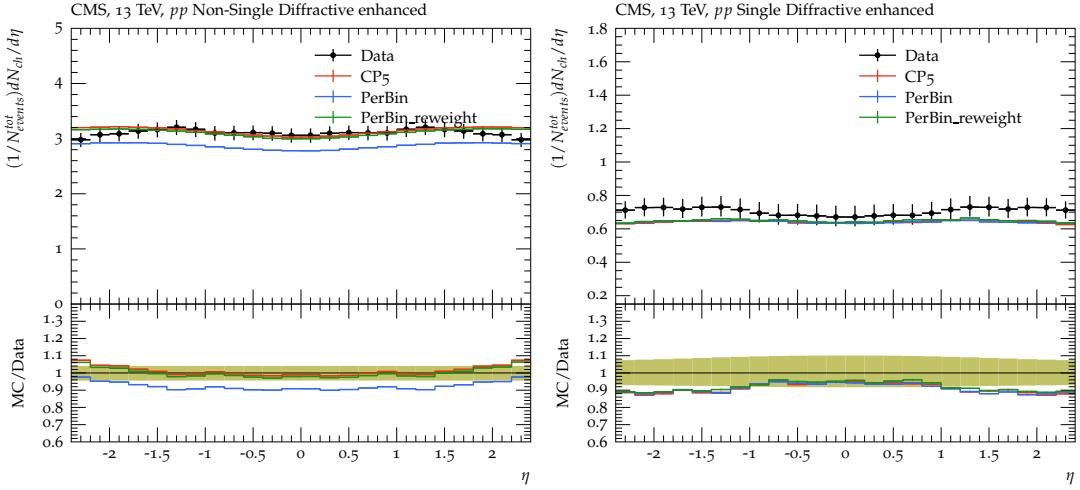
**Figure 6.11:** The transMAX charged particle density (upper left) and the charged  $p_T$ -sum (upper left); the transMIN charged particle density (lower left) and the charged  $p_T$ -sum from the CDF analysis at  $\sqrt{s} = 1.96$  TeV in proton-antiproton collisions [43]. The data are described very 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.

Tune	$\chi^2/\text{DoF}$
CP5:	23.9
PerBin:	13.7
PerBin + re-weights:	19.4

**Table 6.10:**  $\chi^2$  evaluation for the three tunes.

Fig. 6.13. 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. 6.13. This tune in the idea of having a more general tune that describe better all the distributions can be considered as a better tune. In general the  $\chi^2$  evaluated is less in both our tunes respect to the CP5 one.

### 6.3. OUR TUNE FOR THE UE IN MINIMUM BIAS OBSERVATIONS



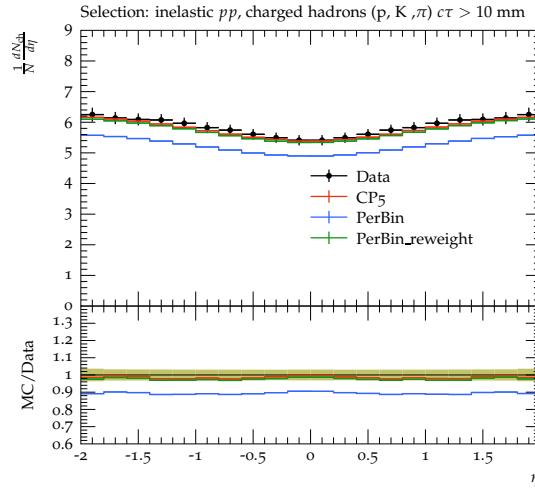
**Figure 6.12:** Here are reported the pseudorapidity distributions ( $p_T > 0.5$  GeV,  $|\eta| < 2.4$ ) for charged particle multiplicity in single diffractive (right) and non-single diffractive (left) events selection. The black point are the data from the CMS analysis at  $\sqrt{s} = 13$  TeV [45]. The data from the NSD events are not so well described from the PerBin model (blue line) but with the re-weights we can describes these data points better. Instead, for the SD events we get a result equal to the one of CP5. 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.

Given these results we can conclude that MCNTUNES is valid tool for the tune of the parameters in high energy physics generators. We obtained two valid tunes for the underlying event and minimum bias observations in proton-proton collisions. The activity observed in the two transverse regions and the pseudorapidity distributions for different events selections are well described after the tune of the parameters that control the multi parton interactions and the color reconnection in PYTHIA8. The output we got from the minimizer in Fig. 6.5 tell to us that the most important parameter for the description of this distributions is the threshold value  $p_{T0}^{ref}$ , all the distributions have a large sensitivity on the variation of this parameter and this is related to the well defined minimum we got.

On the other hand the small dependence on the variation of the `ColorReconnection:-range` parameter over a certain threshold indicate a saturation, in fact over some value almost all the possible color reconnection have taken place and so the model is less sensitive to this parameter.

The Inverse model instead in this case don't give us a good result but maybe further tests can take to some good results. Maybe increasing even more the training set size but with careful, MCNTUNES does not present any control on the NNs typical over-fitting problem. Even if the Inverse Model don't give us a complete tune for the underlying events it perform very well in the first test. So, we cannot exclude it as a valid tool.

Now, that the tool is validated we are going to extend the tune procedure to some new distribution and parameter. In the next chapter we focus on the tune of the



**Figure 6.13:** In this figure is shown the last distribution we use for the tune from the CMS analysis at  $\sqrt{s} = 13$  TeV [42]. The pseudorapidity distribution ( $|\eta| < 2$ ) for the charged hadron density in an inelastic proton-proton scattering selection. Also here the PerBin model (blue line) give 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 distribution the statistical uncertainties are very small due to the high number of events in each bin.

*Primordial  $k_T$  and Space Shower* in order to improve the description of the data collected in  $Z$  boson production events.

# Chapter 7

## Tune for the Primordial $k_T$

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 3.2.7.

The main unresolved problem for the primordial  $k_T$  tune is the unexpected high value required for the description of the observed  $Z$  boson  $p_T$  spectrum.

In fact, the primordial  $k_T$  found is origin in the Fermi Motion of the partons inside the hadrons. So when the parton undergoes to the hard scattering it can already have an initial no zero transverse momentum. The value of the primordial  $k_T$  can be estimate as reported in Eq. 3.29, but in the case of the  $Z$  boson  $p_T$  spectrum this estimation is not sufficient the required value observed in order to reproduce the experimental data are in the order of 2 GeV.

The primordial  $k_T$  was tune in the Monash tune [56] and the following tunes, as CP5, inherited the values of the parameters from there.

So the introduction of a new tune for the primordial  $k_T$  is needed. In fact the CP5 tune it is known that does not describe well the  $Z$  spectrum in the low- $p_T$  region [39]. This is shown in Fig. 7.1 for the production cross-section in  $Z$  plus jets events and in the figure Fig. 7.2 for  $Z$  boson production in DY observations. As we can see the CP5 tune is very far from experimental data values in the low regions.

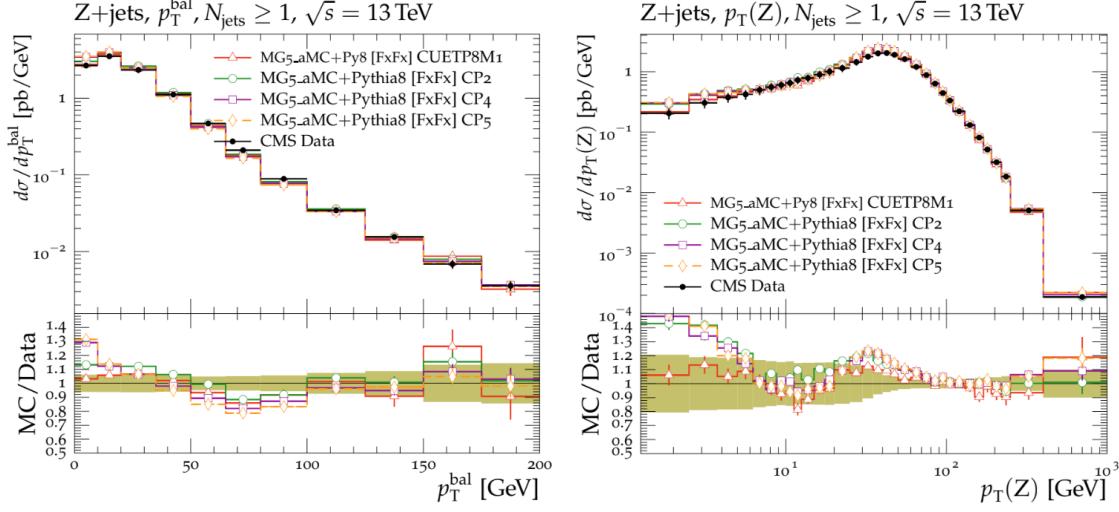
The distribution for  $Z$ +jets events as a function of the variable  $p_T^{bal} = |p_T^Z - \sum_{\text{jets}} p_T^j|$  is not well described in the first bin. While the one as a function of the  $p_T^Z$  miss the description in the region  $p_T^Z \lesssim 40$  GeV. These two observables as described in [39] are very sensitive to the parton shower process and UE.

For the tune we focus on the distribution in Fig. 7.2 distributions, also here we can see that CP5 tune don't describe very well the low regions for these two observables.

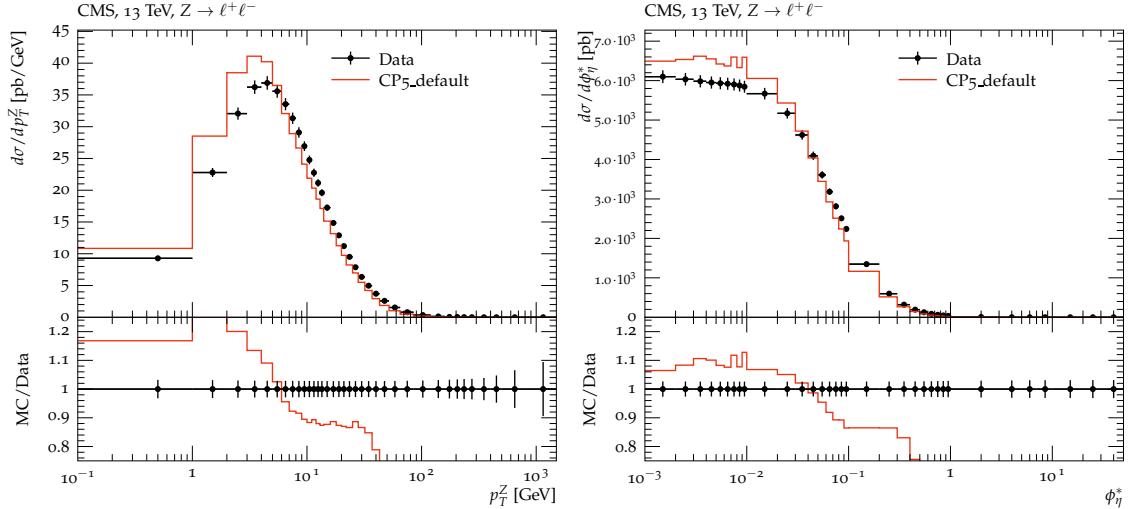
### 7.1 Primordial $k_T$ and ISR effect on $p_T^Z$

The parameters on which we focus in order to tune data and explain the  $Z$  boson  $p_T$  spectrum are:

- `BeamRemnants:primordialKThard` that set the width of the Gaussian distribution for the primordial  $k_T$  sample;
- `SpaceShower:pT0Ref` that set the threshold for the initial state radiation to take place.



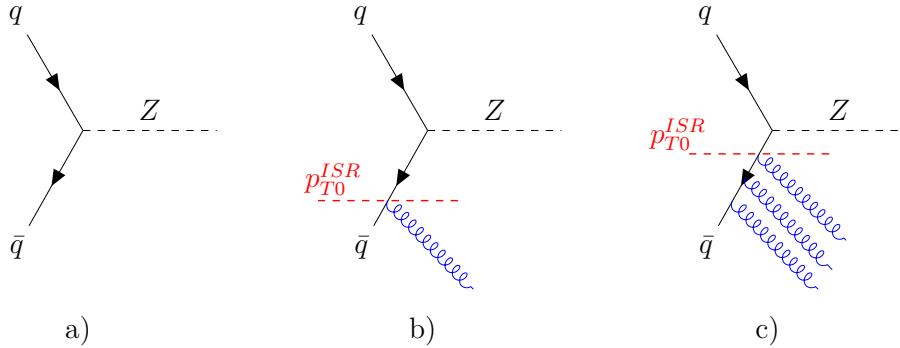
**Figure 7.1:** Figure from CP tunes presentation paper [39]. These two images show the  $Z$  boson production cross-section in  $Z$  plus jets (with at least one jet) at  $\sqrt{s} = 13$  TeV as a function of the imbalance of the transverse momentum between the jet and the  $Z$  boson (left) and of the  $Z$  boson transverse momentum (right) [57]. The first bin in the left distribution and the region  $p_T^Z \lesssim 40$  GeV in the right one are not well described by CP5 tune (yellow line).



**Figure 7.2:** The CP5 tune is not good in the description of the low region of the  $Z$  boson production cross-section of the  $Z$  boson transverse momentum (left) or of the  $\phi_\eta^*$  angle. This distribution are from the CMS analysis [41] at the center-of-mass energy of 13 TeV.

Let's see how this parameter impact on the  $Z$  boson production. If we consider only the LO diagram for the  $Z$  production we can have only a production with a zero transverse momentum. So the  $Z$  spectrum we expect at LO is a  $\delta$ -distribution function.

With the addition of the primordial  $k_T$  we get a Gaussian distribution which width is set by `BeamRemnants:primordialKThard`. If we move 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.

A large contribution come also from 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  $k_T$ .

An higher amount of ISR, so a lower value for the threshold, can lead to a larger  $p_T$  taken from the parton that generate the  $Z$  boson with an higher  $p_T$  and vice versa.

## 7.2 Primordial $k_T$ tune

The primordial  $k_T$  was not investigated by CP5 so it is important to include that in the analysis in order to describe better the low  $p_T$  region.

The parameters variation ranges we use are the one reported in the Table 7.1 while others parameters are set to CP5 values.

Parameter Name	Value
BeamRemnants:primordialKThard	[0.5 – 5.0]
SpaceShower:pT0Ref	[0.5 – 5.0]

**Table 7.1:** Variation ranges for the sampling used in the primordial  $k_T$  tune.

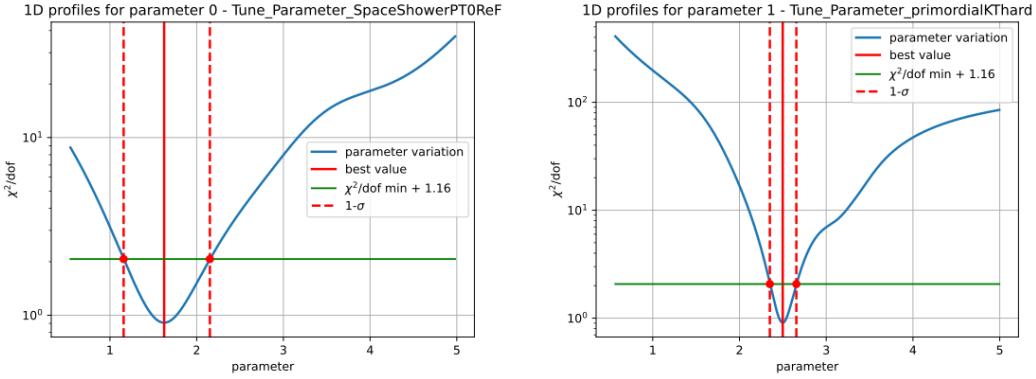
The number of samples is lower than the one used for the Underlying Event in fact the training set we use for the PerBin Model contains more or less 160 MC runs and a little more for the Inverse Model,  $\sim 200$ .

Fig. 7.2 shows the distributions used to performed the tune. The distributions are from the [41] analysis at  $\sqrt{s} = 13$  TeV:

- The  $Z$  boson production cross-section in DY observation as a function of  $p_T^Z$ ;
- The  $Z$  boson production cross-section in DY observation as a function of  $\phi_\eta^*$ .

For the tune we use only the

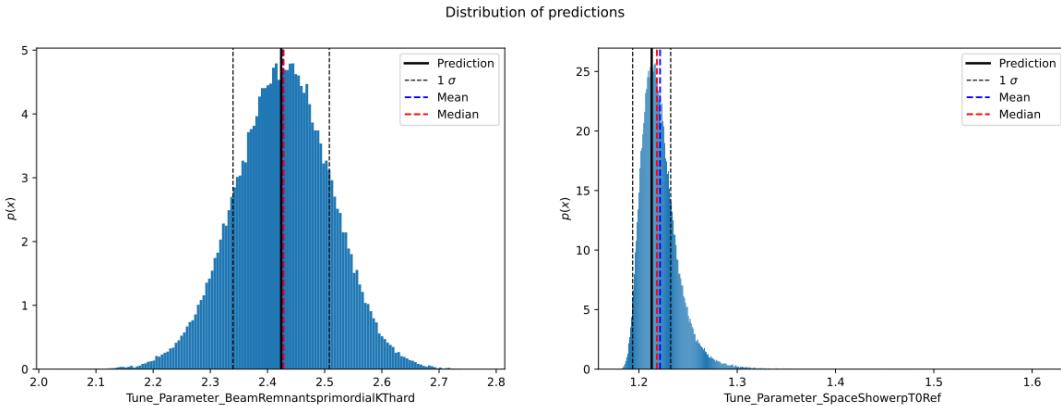
The PerBin Model minimization results are displayed in Fig. 7.3. The blue line indicates the value of the  $\chi^2/DoF$  as a function of the parameter value. The best



**Figure 7.3:** 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  $\chi^2/\text{DoF}$  the best value is indicated by the solid red line while the errors are the ones indicated by the dashed red lines.

estimation is marked by a solid red line while the dashed red lines indicate the errors. Both the parameters are in a well defined minimum.

The Inverse model distribution of prediction for the best configuration is reported in Fig. 7.4 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 hyperparameters space scanned to search for the best configuration is the same used above and reported in Table 6.3. The best hyperparameters we found are the one reported in the Table 7.2



**Figure 7.4:** The prediction spread for the Inverse model in the primordial  $k_T$  tune. The left histograms 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.

The PerBin Model gives to us very stable results the value and the errors we got from the tune are reported in Table 7.3. This are also compared to the ones obtained from the Inverse model. The value obtained from the two model are compatible each other. The default values for CP5 are also reported and they are the one inherited

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

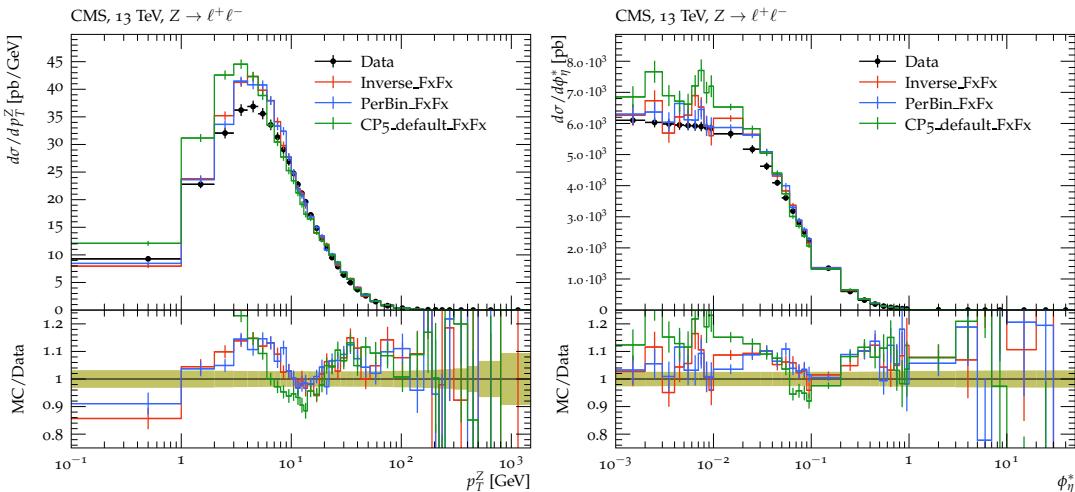
**Table 7.2:** Best hyperparameters model found for the primordial  $k_T$  tune with Inverse model.

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

**Table 7.3:** 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.)

by Monash13 tune. As we can see the value we get are quite different to the one in CP5.

The overall results are displayed in Fig. 7.5. Our tunes describe better the low region of the two spectrum. These low regions are the one we actually tune 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. 7.5. To simulate all the spectrum we have to use a margin scheme as FxFx in order to avoid the double counting.

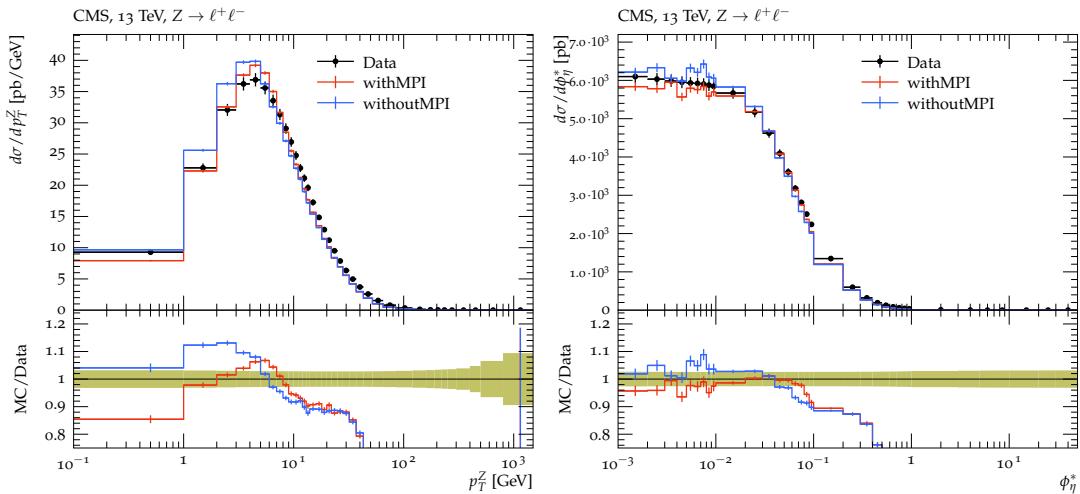


**Figure 7.5:** The results we get from the tune of the primordial  $k_T$ . The left figure shows the  $Z$  boson production cross-section as a function of the  $p_T^Z$ . The red line refers to the PerBin model tune, the blue line to the Inverse Model and the green line is the CP5 default tune. The black points are the experimental data. The colored vertical lines the statistical uncertainties. It is clear that our two tunes better describe the low region respect to the original CP5.

### 7.3 Primordial $k_T$ tune vs MPI

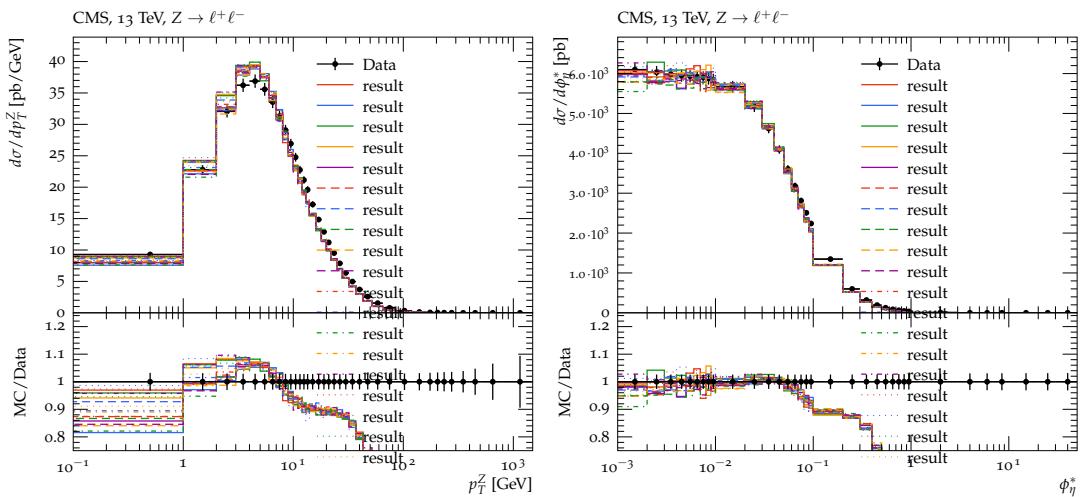
As discussed above primordial  $k_T$  is not the only source of non zero transverse momentum in a LO  $Z$  boson production. This is the reason why we are tuning also the Initial State Radiation. In fact the ISR can leads to a non zero initial  $k_T$  a larger amount of ISR (low `SpaceShower:pT0Ref`) is related to more splitting occurring in the evolution of the incoming partons and before the hard scattering in which the  $Z$  boson is produced. But this is not the only one process that can leads to a non-zero initial transverse momentum, there are also the MPI that can contribute to this initial transverse momentum.

Fig. 7.6 shows the effect of the MPI on the distributions we use for the tune. The blue line is obtained setting `PartonLevel:MPI=off` in PHOTIA8 configuration.



**Figure 7.6:** The image shows that MPI have an effect on the two distribution. The blue line is the result without the MPI and the red line with MPI.

The main parameters 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. 7.7 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 7.7:** The image shows that MPI have an effect on the two distribution. The blue line is the result without the MPI and the red line with MPI.



# Conclusions

In this work it has been validate MCNNTUNES as a valid tool for the tuning of the high energy physics Monte Carlo generator PYTHIA8. Firstly, in order to validate it, the underlying event was tuned trying to reproduce the results obtained with CP5. The work started with a test phase with the variation of only two parameters: `MultipartonInteractions:pT0Ref` and `MultipartonInteractions:ecmPow`. Both the MCNNTUNES models perform very well in this test phase.

After, this test the number of parameters was extended to the same parameters tuned in CP5. So, to the two parameters already tuned have been added the parameters `MultipartonInteractions:coreRadius`, `MultipartonInteractions:coreFraction` and `ColorReconnection:range`. In this phase we obtain very good results from the PerBin model. It seems to describe data also better then CP5 but not in all the distributions, in particular the distributions with larger experimental uncertainties. So, it was decided to include a re-weight for some bins. This leads to overall results more similar to the CP5 ones.

On the other hands the Inverse model gives very bad results in this case it is not able to produce some consistent results. This can be related to the complexity operation that is performed by this model: learn the function that describe the generator respond and try to invert it. This operation is very complex and this complexity increase 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 on these distribution of the parameters `BeamRemnants:-primordialKThard` and `SpaceShower:pT0Ref` was investigated. In fact, the CP5 tune does not describe these distribution very well, the values for these two parameters were inherited by the older Monash Tune. The new tune presented here for these two parameters was performed with both the models offered by MCNNTUNES and the results obtained are compatible each other.

In the future works also the impact of the MPI has 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 can become a cross check to the already existing and consolidate tool PROFESSOR. Moreover, if some more attention is given to the Inverse model it can become a new powerful approach to the tuning procedure.



# Appendix A

## Test with two free parameters for the Underlying Events

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.

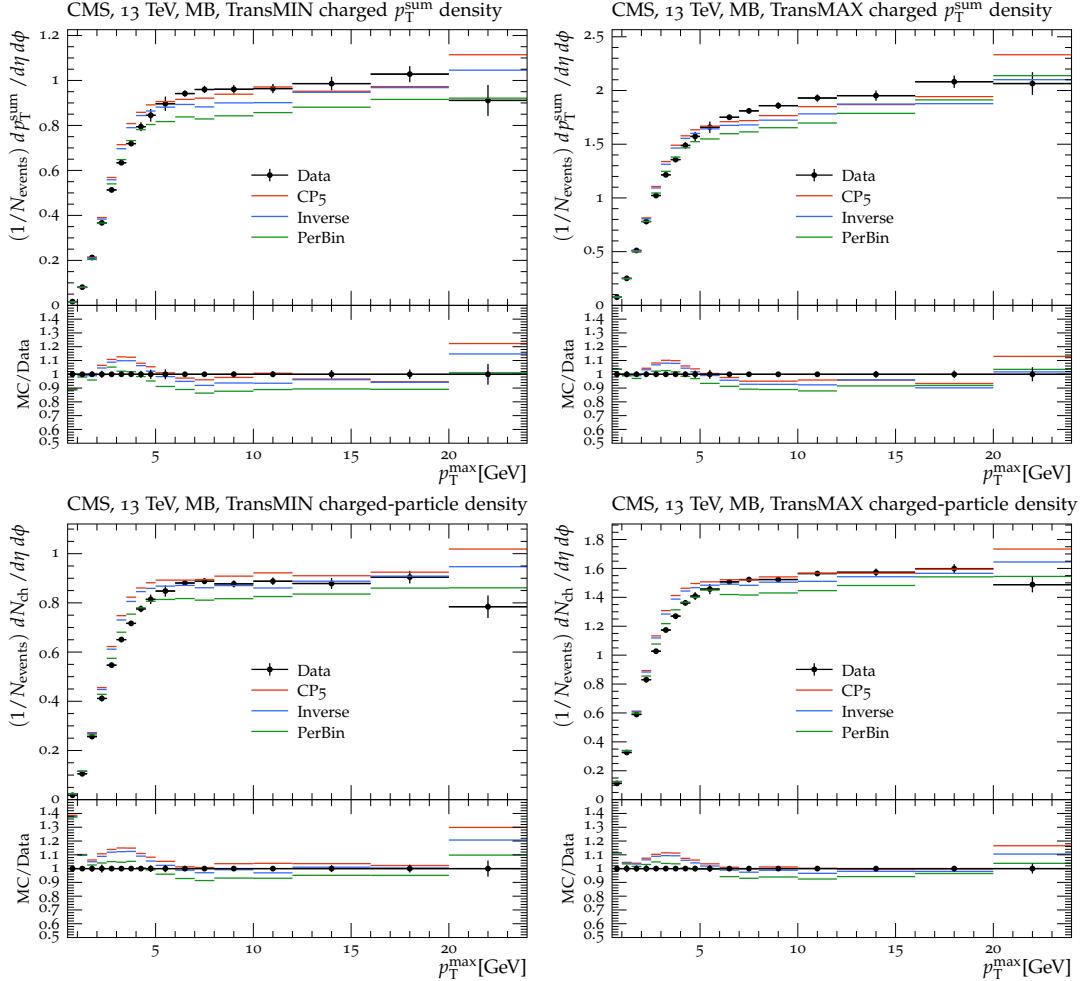
The value we get are also reported in the Table A.1. The PerBin Model errors are not reported because there was not a proper errors estimation correctly implemented yet for this model.

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

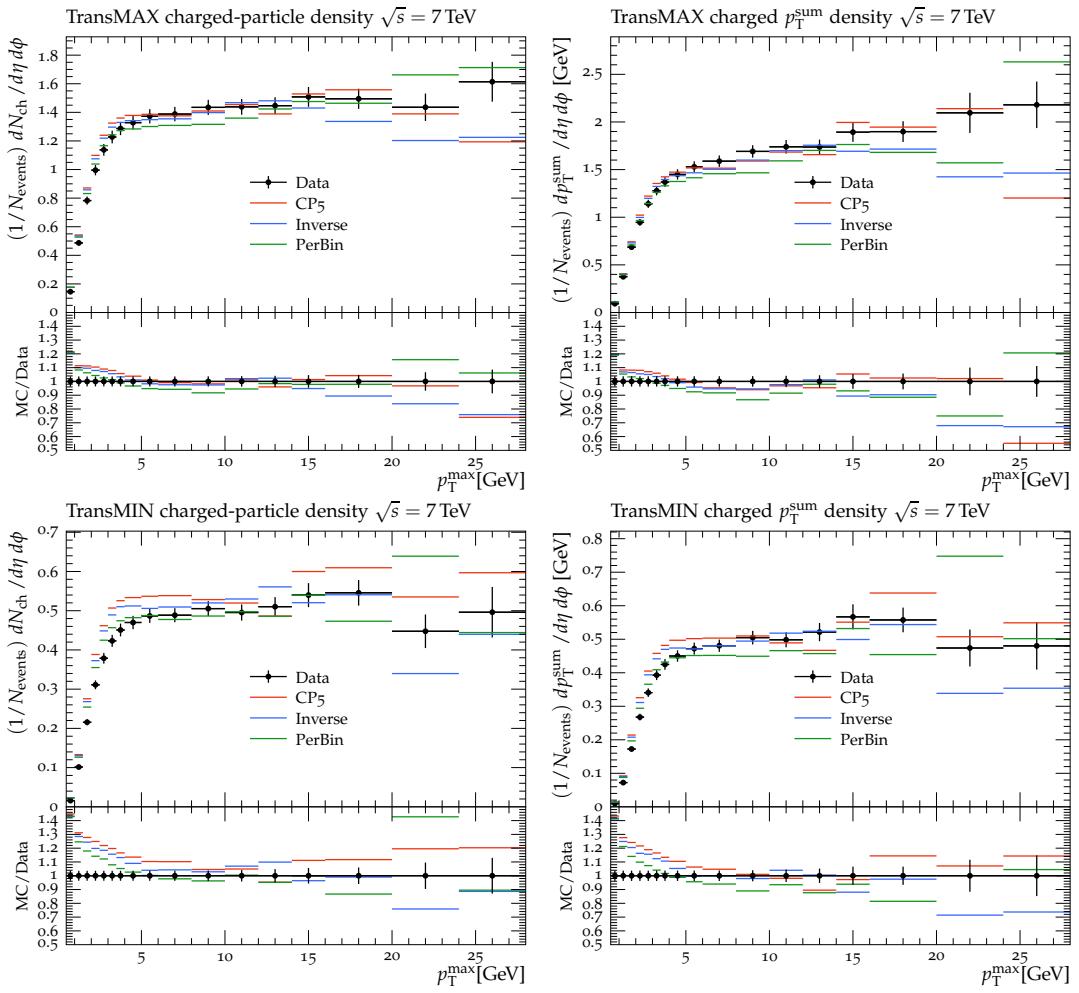
**Table A.1:** Result PerBin model and Inverse Model in two parameter variation test. The value are compared to the upper and lower limit for CP5.

APPENDIX A. TEST WITH TWO FREE PARAMETERS FOR THE  
UNDERLYING EVENTS

---



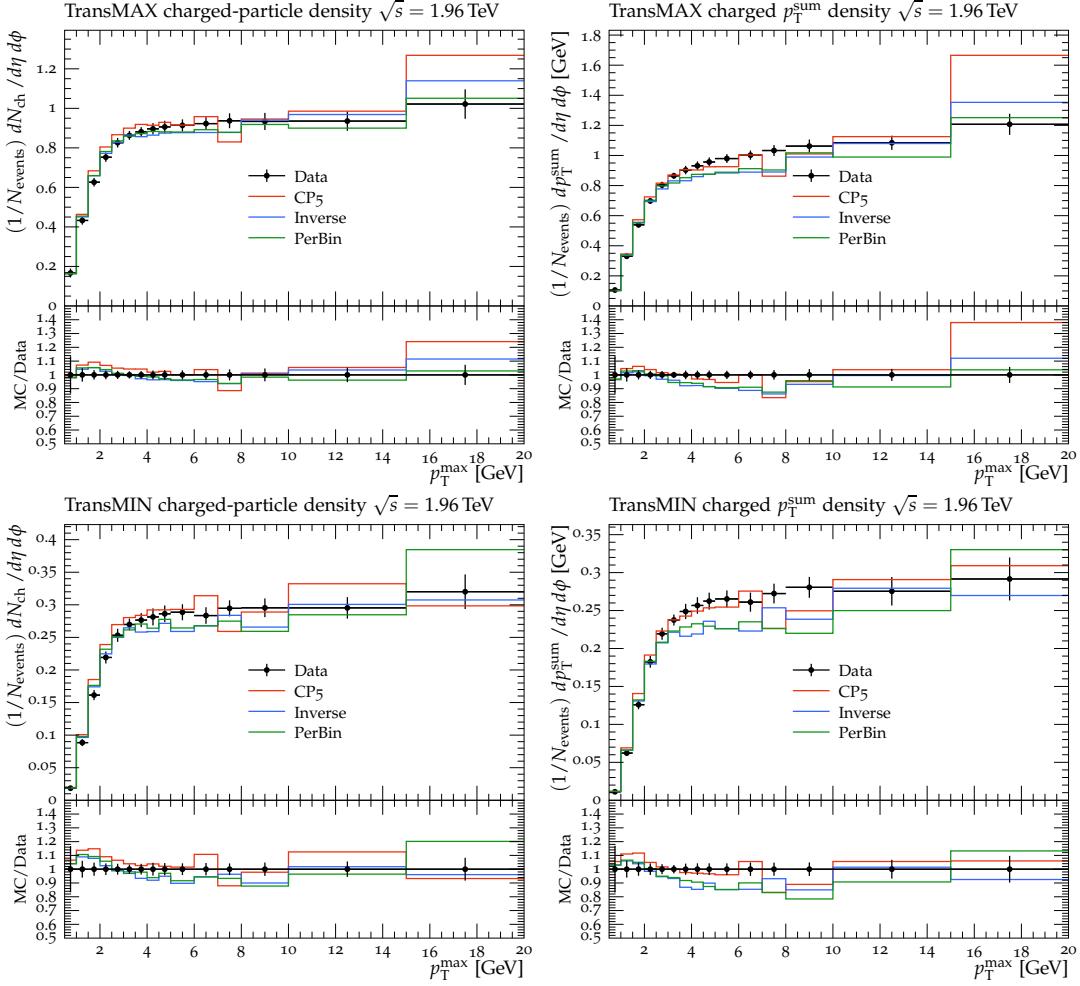
**Figure A.1:** In this figure the data from the  $\sqrt{s} = 13$  TeV CMS analysis [34] that show the transMAX charged particle density (upper left) and the charged  $p_T$ -sum density (upper right); the transMIN charged particle density (lower left) and the charged  $p_T$ -sum density (lower right) 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.



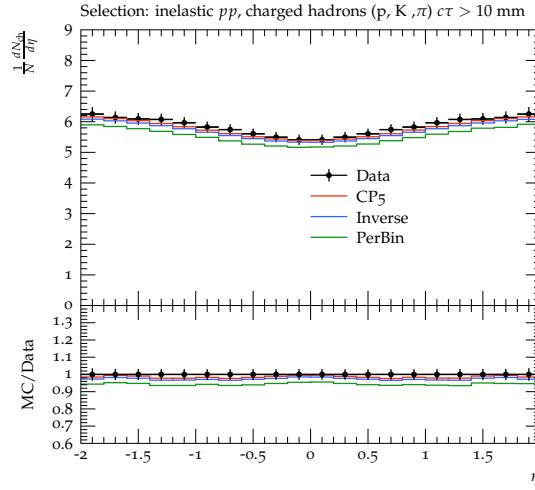
**Figure A.2:** Here the data at  $\sqrt{s} = 7$  TeV from the CMS analysis [44] for the transMAX charged particle density (upper left) and the charged  $p_T$ -sum (upper right); the transMIN charged particle density (lower left) and the charged  $p-T$ -sum are displayed as a function of the transverse momentum of the leading object.

APPENDIX A. TEST WITH TWO FREE PARAMETERS FOR THE  
UNDERLYING EVENTS

---



**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 [43].



**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 [42]. 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] Giovanna Francesca Cottin Buracchio. Phenomenology of new physics beyond the Standard Model: signals of Supersymmetry with displaced vertices and an extended Higgs sector at colliders, Aug 2017. URL <https://cds.cern.ch/record/2276806>. Presented 09 Jun 2017.
- [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] P.A. Zyla et al. Review of Particle Physics. *PTEP*, 2020(8):083C01, 2020. doi: 10.1093/ptep/ptaa104. and 2021 update.
- [6] J. D. Bjorken. Asymptotic Sum Rules at Infinite Momentum. *Phys. Rev.*, 179: 1547–1553, 1969. doi: 10.1103/PhysRev.179.1547.
- [7] 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>.
- [8] 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>.
- [9] L N Lipatov. The parton model and perturbation theory, 1975. URL <http://cds.cern.ch/record/400357>.
- [10] Vladimir Naumovich Gribov and L N Lipatov. Deep inelastic ep scattering in perturbation theory, 1972. URL <https://cds.cern.ch/record/427157>.
- [11] G. Altarelli and G. Parisi. Asymptotic freedom in parton language, 1977. ISSN 0550-3213. URL <https://www.sciencedirect.com/science/article/pii/0550321377903844>.
- [12] Yuri L. Dokshitzer. Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics., 1977.

## BIBLIOGRAPHY

---

- [13] W.J. Stirling. private communication. URL <http://www.hep.ph.ic.ac.uk/~wstirlin/plots/plots.html>.
- [14] 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>.
- [15] Toichiro Kinoshita. Mass singularities of feynman amplitudes, 1962. URL <https://doi.org/10.1063/1.1724268>.
- [16] T. D. Lee and M. Nauenberg. Degenerate systems and mass singularities, Mar 1964. URL <https://link.aps.org/doi/10.1103/PhysRev.133.B1549>.
- [17] A. Kulesza, G. Sterman, and W. Vogelsang. Electroweak vector boson production in joint resummation, 2002. URL <https://arxiv.org/abs/hep-ph/0207148>.
- [18] 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>.
- [19] Manuel Bähr, Stefan Gieseke, Martyn A. Gigg, David Grellscheid, Keith Hamilton, Oluseyi Latunde-Dada, Simon Plätzer, Peter Richardson, Michael H. Seymour, Alexander Sherstnev, and Bryan R. Webber. Herwig++ physics and manual, Nov 2008. ISSN 1434-6052. URL <http://dx.doi.org/10.1140/epjc/s10052-008-0798-9>.
- [20] T Gleisberg, S Hoeche, F Krauss, A Schaelicke, S Schumann, and J Winter. Sherpa 1. , a proof-of-concept version, Feb 2004. ISSN 1029-8479. URL <http://dx.doi.org/10.1088/1126-6708/2004/02/056>.
- [21] E. Boos, M. Dobbs, W. Giele, I. Hinchliffe, J. Huston, V. Ilyin, J. Kanazaki, K. Kato, Y. Kurihara, L. Lonnblad, M. Mangano, S. Mrenna, F. Paige, E. Richter-Was, M. Seymour, T. Sjostrand, B. Webber, and D. Zeppenfeld. Generic user process interface for event generators, 2001.
- [22] Stefano Catani, Frank Krauss, Bryan R Webber, and Ralf Kuhn. Qcd matrix elements + parton showers. *Journal of High Energy Physics*, 2001(11):063–063, Nov 2001. ISSN 1029-8479. doi: 10.1088/1126-6708/2001/11/063. URL <http://dx.doi.org/10.1088/1126-6708/2001/11/063>.
- [23] 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>.
- [24] 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>.

- [25] Stefano Frixione and Bryan R. Webber. The mc@nlo 3.1 event generator, 2005.
- [26] Rikkert Frederix and Stefano Frixione. Merging meets matching in MC@NLO. *JHEP*, 12:061, 2012. doi: 10.1007/JHEP12(2012)061.
- [27] Victor S. Fadin. BFKL resummation. *Nucl. Phys. A*, 666:155–164, 2000. doi: 10.1016/S0375-9474(00)00022-1.
- [28] I. I. Balitsky and L. N. Lipatov. The Pomeranchuk Singularity in Quantum Chromodynamics. *Sov. J. Nucl. Phys.*, 28:822–829, 1978.
- [29] 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>.
- [30] FRANK SIEGERT. Monte-carlo event generation for the lhc, 2010. URL <http://etheses.dur.ac.uk/484/>.
- [31] 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](https://doi.org/10.1016/0370-1573(83)90080-7). URL <https://www.sciencedirect.com/science/article/pii/0370157383900807>.
- [32] Torbjorn Sjostrand. Jet Fragmentation of Nearby Partons. *Nucl. Phys. B*, 248: 469–502, 1984. doi: 10.1016/0550-3213(84)90607-2.
- [33] 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.
- [34] 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>.
- [35] 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.
- [36] 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.
- [37] 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.
- [38] Florian Bechtel. *The underlying event in proton-proton collisions*. PhD thesis, Hamburg U., 2009.

## BIBLIOGRAPHY

---

- [39] 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.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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>.
- [45] Albert M. Sirunyan et al. Measurement of charged particle spectra in minimum-bias events from proton–proton collisions at  $\sqrt{s} = 13$  TeV. *Eur. Phys. J. C*, 78(9):697, 2018. doi: 10.1140/epjc/s10052-018-6144-y.
- [46] 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.
- [47] Stefano Carrazza and Marco Lazzarin. N3pdf/mcnntunes: mcnntunes 0.1.0, October 2020. URL <https://doi.org/10.5281/zenodo.4071125>.
- [48] 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.
- [49] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 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](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](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] 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.
- [57] 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.