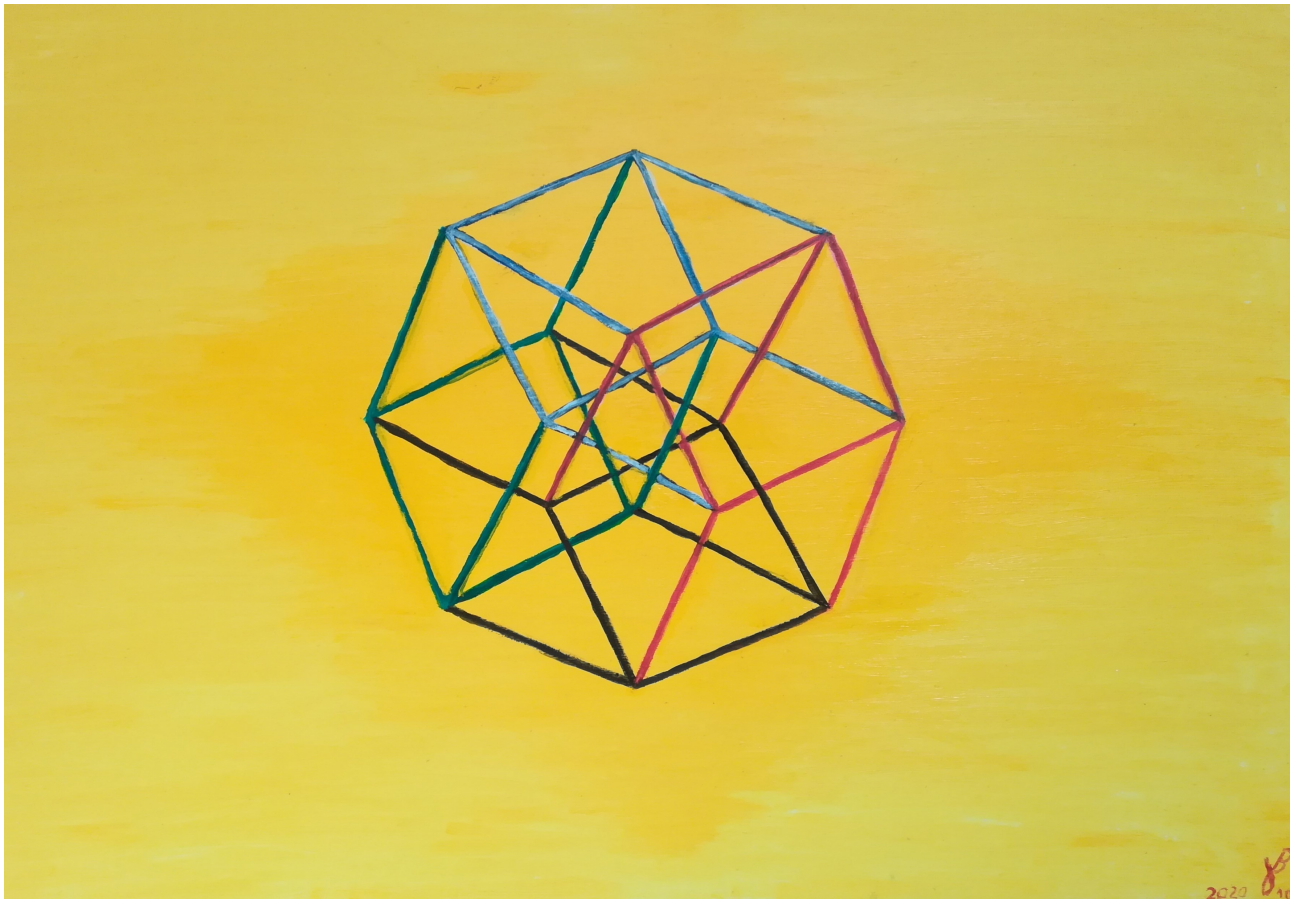
UNIVERSITÀ DEGLI STUDI DI PAVIA
DOTTORATO DI RICERCA IN FISICA – XXXIII CICLO

Phenomenology of Transverse Momentum Distributions in hadronic observables

Chiara Bissolotti



Tesi per il conseguimento del titolo

Università degli Studi di Pavia
Dottorato di Ricerca in Fisica - XXXIII Ciclo

UNIVERSITÀ
DI PAVIA

Istituto Nazionale di Fisica Nucleare

Phenomenology of Transverse Momentum Distributions in hadronic observables

Chiara Bissolotti

Submitted to the Graduate School of Physics in partial fulfillment of the requirements for the degree of

Dottore di Ricerca in Fisica

Doctor of Philosophy in Physics

at the

University of Pavia

Supervisor: Prof. Alessandro Bacchetta

Cover: "Hypercube and multidimensions: keys to understand energy?", *oil painting on canvas*, by Giorgio Bissolotti

Phenomenology of Transverse Momentum Distributions in hadronic observables

Chiara Bissolotti

PhD Thesis - University of Pavia
Pavia, Italy, May 2021

Alla mia famiglia

Contents

Abstract

One of the main aims of hadronic physics is to describe the internal structure of the nucleon in terms of its constituents, quarks and gluons (collectively called partons). A lot of information has been collected over the past forty years concerning the distribution of partons in one dimension, encoded in the well–known collinear Parton Distribution Functions (PDFs). In the last years, we are extending the study to the distribution of partons in full three–dimensional momentum space, encoded in the so–called Transverse Momentum Distributions (TMDs). This thesis describes a suite of computing tools for the extraction of TMDs, entirely developed by our research group over the last three years, and presents a state–of–the–art extraction of these functions from experimental data.

The thesis first summarizes the theoretical framework for the extraction of TMDs in two types of scattering processes: the Drell–Yan process ($pp \to l\bar{l}X$), and Semi—Inclusive Deep Inelastic Scattering ($lp \to lhX$).

The thesis then presents the numerical framework we implemented to study TMDs. It consists of a suite of tools, which we called `NangaParbat`. It is written in C++ and is publicly available. `NangaParbat` can be used to extract TMDs, to produce grids for TMDs and TMD–related observables, and to have easy access to TMD extractions, through interpolation and convolution tools. Therefore, NangaParbat can be a very useful asset for the scientific community working on the phenomenology of hadronic physics.

Finally, the thesis presents our most recent extraction of TMDs, which reached the unprecedented accuracy of Next–to–Next–to–Next–to–Leading Logarithm (N3LL). We used Drell–Yan data from various experiments, including those at the LHC, and spanning a wide kinematic range. We obtained a very good description of both the shape and the normalization of the data without introducing normalization coefficients as it was done in the literature before: this result was made possible only through the unsurpassed perturbative accuracy of the fit and the optimized numerical and analytical integration techniques that we used.

Chapter 1

Introduction

There are four different forces observed in nature: the strong force, the electromagnetic force, the weak force and the gravitational force. This thesis deals with topics related to the strong force, the force that determines the structure of protons and neutrons and keeps them together in the nucleus. All particles that are subject to the strong force are called hadrons (from the Greek ἁδρός, strong) and the field of physics that studies the strong force is called hadronic physics.

In 1964 Gell–Mann (1969 Nobel Prize) [1] and George Zweig [2], independently, proposed that many properties of hadrons could be explained assuming they were made of "quarks." [1] A revolution in physics was underway, and it was later realized that quarks come in three copies, which differ from each other only for a property that was called "color charge", or simply "color". In other words, through the property of color we describe the three different quantum states that quarks can exist in: in 1964, Oscar W. Greenberg introduced the notion of color charge to explain how quarks could coexist inside some hadrons in otherwise identical quantum states without violating the Pauli exclusion principle.

The forces between quarks are due to the property of color: the strong interaction is mediated by the exchange of massless particles called gluons (because they "glue" quarks together) that carry the "color" charge and interact with quarks, antiquarks, and other gluons.

Nature forces quarks to be always bound in colorless states, the hadrons, and therefore is not possible to observe a quark in its free state. This phenomenon is known as color confinement, or simply *confinement*, and, even if it has been known for several decades by now, there is no mathematical proof that explains it yet.

At first, it was not clear whether quarks were real particles or just mathematical

[1] The term "quark" was first adopted by Gell–Mann, who borrowed it from the sentence "Three quarks for Muster Mark" in James Joyce's *Finnegans Wake*.

tools useful to understand the properties of hadronic particles, especially since the strong interaction had yet to be properly enclosed in a suitably formulated theory.

In order to investigate the dynamics obeyed by quark systems, it seemed natural to probe the inside of hadrons (protons, for example) by applying a beam of point–like particles, such as electrons, that act like a "sharp knife" and cut cleanly through hadronic matter revealing its inner structure.

The first runs of this kind of experiments, to test the structure of the proton, started in the early 1960s and were carried out at high–energy electron accelerators, like the Stanford Linear Accelerator (SLAC), and the process was called Deep Inelastic electron–proton Scattering (DIS). Then, an experiment performed by the SLAC–MIT collaboration in 1967 [3] gave results that could be interpreted as direct evidence of the physical existence of quarks [4, 5].

In 1969, Bjorken [6] noted that the functions that express the partonic structure of hadrons, called *structure functions*, enjoyed a particular property: in the deep inelastic region, where the momentum transfer squared Q^2 and the energy transfer ν of leptons are very large, structure functions in electron–nucleon scatterings depend only on the ratio Q^2/ν rather than on the two independent variables Q^2 and ν. In high–energy scattering experiments, a property of hadrons is said to scale when it is determined not by the absolute energy of an experiment but by dimensionless kinematic quantities, such as a scattering angle or the ratio of the energy to a momentum transfer. The claim that the property of scaling of structure functions was expected in the deep inelastic region, where the ratio Q^2/ν is constant, was later called *Bjorken scaling*.

One of the simplest ways to understand Bjorken scaling was to think that the leptons of the beam collide against almost free point–like particles inside the nucleons: this was the idea behind the *parton model*, proposed in 1969 by Richard Feynman [7]. Nucleon constituents in this model were called *partons*: they were later identified with quarks and gluons, since experiments demonstrated that they had the same quantum numbers such as charges and spins.

After the first proposals of the existence of quarks, their freedom inside nucleons and, at the same time, their property of color confinement, all the quantum theories known at the time were examined for compatibility with the requirements of quarks dynamics. This thorough and lengthy process led to the knowledge that a theory that describes the interaction of quarks and gluons has to belong to the class of non–Abelian gauge field theories [8]. The non–Abelian character is reflected in the fact that gauge bosons themselves are carrying color charges: as a consequence, gluons are self–coupling.

In 1973 it was shown that non–Abelian gauge theories can display the property of *asymptotic freedom*: the forces between quarks become asymptotically weaker as the energy increases. This was was a breakthrough that made it possible to formulate a

Quantum Field Theory of strong (colored) interactions, called Quantum Chromodynamics (or, in short, QCD), with point–like behavior of quarks at short distance and the strong confining force at large distance.

Non–Abelian gauge fields are characterized by symmetries described by a non–commutative algebra: this implies that quarks must have an extra symmetry associated with the non–Abelian gauge field describing the quark dynamics. In 1973, Fritzsch and Gell–Mann [9] suggested that this extra symmetry was to be identified with the color symmetry: with such recognition, most of the properties of quark systems were explained in a natural way and QCD was finally established as the theory of strong interactions between quarks and gluons, the force carriers of the theory.

If we were to draw a comparison between QCD and QED (Quantum Electrodynamics), the Abelian gauge theory that describes electromagnetic interactions, we could say that color is the QCD analog of the electric charge in QED. One of the main differences between QCD and QED, though, is that while photons, the mediators of electromagnetic interactions, do not carry any electric charge, and thus can not interact with each other, in QCD gluon–gluon interactions are possible, since also gluons are colored.

Asymptotic freedom, which was worth the Nobel prize in 2004 [10], allows perturbative treatment of strong interactions at short distances. The method of using perturbation theory in QCD is often referred to as *perturbative QCD* (or *pQCD* for short).

Investigations of quark dynamics started right after the foundation of the parton model. Deep inelastic scattering was the first process to which QCD was applied, in order to provide an explanation of Bjorken scaling: this involves separating the purely short–distance part out of the deep inelastic cross section, to assure that perturbative rules are used in their proper regime of application. The operator product expansion (OPE) [11] was the most powerful known tool to accomplish this task, and with it one finds that the lowest–order result in the perturbative calculations reproduce Bjorken scaling. In this way, the parton picture was given a solid theoretical foundation.

However, these studies also showed that the perturbative treatment is spoiled by large logarithms, associated with the presence of a large momentum transfer in the process: because of this, it is necessary to sum those large logarithms to all orders using renormalization group equations (RGE).

Because of this, it was discovered that Bjorken scaling holds only in an approximate way, namely, only at what is called *leading order*. The deviation from the Bjorken scaling in structure functions was later confirmed by experiments of muon–nucleon scattering such as the NA2 experiment of the European Muon Collaboration (EMC) [12, 13] and the NA4 experiment of the Bologna–CERN–Dubna–Munich–Saclay (BCDMS) Collaboration [14, 15], and these observations confirmed that QCD was a valid theory (see also the review paper [16]).

QCD was promptly applied also to the process of electron–positron annihilation. In this process is possible to define observables that are free from long–distance effects and can be computed directly using directly perturbation theory. Perturbative calculations of the total cross section of e^+e^- annihilation, using RGE, began right after the discovery of asymptotic freedom [17].

In 1977 Altarelli and Parisi published a paper [18], with the title *"Asymptotic Freedom in Parton Language"*, in which they proposed a new way to describe partons and their interactions: this formalism conveys a clear physical meaning, since it directly involves parton densities, defined in the deep inelastic region and at all orders of the running coupling constant for QCD, α_s, called parton distribution functions (PDFs) [19, 20] and parton fragmentation functions (FFs) [21, 22]. Later, it was realized that also Dokshitzer [23] and, independently, (and even a few years earlier, in 1972) Gribov and Lipatov [24] did derive the same equations of the Altarelli–Parisi formalism, and QCD evolution equations were named after the initial of all their authors, DGLAP.

In 1977 a second phase for developments in perturbative QCD started. Three main events were particularly significant and widened the horizon of QCD applicability: the completion of the calculation of next–to–leading–order (NLO) contributions in deep inelastic scatterings [25–27], the introduction of a new type of processes, *i.e.*, jets from quarks and gluons [28–30], and the generalization of OPE to include a larger class of short–distance processes [31–34]. Higher order terms for hadronic observables for many processes (for example, among the others, e^+e^- annihilation [35–39], photon–photon scattering [40], jets in e^+e^- annihilations [41, 42]) have been computed as well.

When dealing with higher order effects, it is immediate to realize that results crucially depend on the method used to renormalize the diverging integrals that show up in the calculations: this is known as *renormalization scheme dependence* of perturbative predictions, and is a common issue in the perturbative treatment of renormalizable quantum field theories.

In fact, renormalization scheme dependence in perturbative predictions occurs also in quantum electrodynamics (QED): in this case, though, since the coupling constant (the expansion parameter in perturbation series) is very small ($\alpha = e^2/4\pi \sim 1/137$), the effect of different choices of renormalization schemes is negligible. On the contrary, in QCD α_s is a running coupling constant and can assume quite large values for a perturbative expansion parameter, making the effects of renormalization scheme dependence much more evident.

In a renormalizable perturbation series, order by order any physical quantity is a function of three classes of variables with dimensions of mass: the kinematic energy scale (or scales) of the scattering Q, the masses m, and a renormalization scale μ. There is a certain freedom in the choice of the renormalization scale μ. However, μ

will appear in ratios Q/μ and μ/m, and at high energy at least one of these ratios is large. As a result, the perturbation series is no longer an expansion in a small parameter: for example, if we choose $\mu \sim Q$, the effective coupling constant g(μ) will generally appear in the combinations $g^{2n}(Q)ln^n(Q/m)$ or $g^{2n}(Q)ln^{2n}(Q/m)$, where n is an integer number related to the perturbative order.

In general, a cross section is a combination of short– and long–distance behavior, and is hence not computable directly in perturbation theory for QCD. There are, however, a few classes of processes for which it is possible to demonstrate factorization theorems that allow us to derive predictions for cross sections by separating (factorizing) long–distance from short–distance behavior in a systematic way. In other words, factorization theorems enable one to apply perturbative calculations to some important processes involving hadrons (in the following denoted with h_1, h_2 or h): for example, factorization is expected to hold for the Drell–Yan process $NN \to l^+l^-X$ (where N is a nucleon, $l^\pm$ charged leptons and X unobserved hadrons), Deeply Inelastic Scattering $lh \to l'X$, $h_1h_2 \to jet\,X$, and inclusive cross sections in electron–positron annihilation with detected hadrons, $e^+e^- \to h + X$.

In Ref. [43] Collins, Soper and Sterman give explicit factorization theorems for the cross sections of deeply inelastic scattering, single–particle inclusive annihilation and the Drell–Yan process. Factorization is also discussed in Chapter 8 of Ref. [44], in Ref. [45] and references therein.

Since 1978, a variety of processes, such as, among others, the Drell–Yan process [46, 47], inclusive e^+e^- annihilation [48, 49], $e^+e^- \to$ hadron $+X$ have been analyzed.

However, although many phenomenological studies have been performed since the birth of QCD and successful comparisons with data have been made for a variety of hadronic observables, the structure of the most common of hadrons, the nucleon, is still largely unknown, in particular in its spin–dependent aspects. This is partly due to the fact that in the kinematic regions where the value of α_s is too big to ensure a safe application of perturbative methods, a significant nonperturbative input is required.

Studies of hard scattering processes whose initial state involves nucleons allow us to gain information on their internal structure: as already briefly mentioned above, such knowledge is encoded in parton distribution functions (PDFs).

Typically, in a high–energy collision, there is a direction along which the momentum of the incident and scattered particles is significantly bigger: the longitudinal direction, the one of the beam pipe along which particles are accelerated. This is the direction considered in PDFs: collinear PDFs describe the structure of the nucleon as a function of the nucleon longitudinal momentum fraction x carried by a parton of a certain species.

Unpolarized collinear PDFs are indispensable building blocks in almost any prediction involving high-energy hadrons, and after decades of studies they are fairly

well–known objects. They are mainly studied in proton–proton collisions and Deeply Inelastic Scattering, a process where a lepton l scatters on a proton p that breaks off in other undetected objects X ($lp \to l'X$): here the longitudinal momentum structure of the nucleon is investigated using the collinear approximation, which means that transverse degrees of freedom are not taken into account. Thanks to collinear factorization theorems, which allow to express physical measurements in terms of hard, perturbative subprocesses and universal (measurable) parton distributions, it is possible to give a definition of collinear PDFs [50–52].

1.1 Transverse Momentum Distributions

As discussed above, the proton is a composite particle, and this means that its constituents, quarks and gluons, have their own intrinsic momenta, which in general have both longitudinal and transverse components (the transverse components are the two orthogonal directions perpendicular to the longitudinal direction). The transverse momentum components of partons are still present even when the proton is accelerated along the longitudinal direction and they have to be taken into account for a complete description of the process: the transverse momentum of partons is a central element in understanding the three–dimensional structure of the nucleon. Moreover, transverse components are needed not only for a correct description of the nucleon, but also because they have experimentally observable effects.

In order to accurately describe the proton internal structure, it is necessary to examine more than only one dimension, and hence the interest in the study of Transverse Momentum Dependent Parton Distribution Functions (TMD PDFs, or more concisely, TMDs), which represent three–dimensional maps of the proton in the momentum space (they represent the distribution of both x and the two component of the transverse momentum, $k_\perp$)(see, *e.g.*, Refs. [53–55] and references therein).

In the operator product expansion of the scattering amplitude, the twist is used to classify the operators and isolate the leading (twist–2) from the subleading (higher–twist) contributions, which are power suppressed as $Q^{2-\text{twist}}$. At leading twist, taking into account the intrinsic transverse momentum of the quarks $k_\perp$, eight TMDs are needed to fully describe the nucleon [56–59]: as mentioned before, by adding transverse momentum dependence to collinear PDFs, sensitivity to additional polarization configurations is introduced, and so there are different combinations to consider, as indicated in Tab. 1.1. There are eight quark TMDs because that is the number of parameterizations of the correlator that remain after imposing constraints of hermiticity, parity and time reversal. For more details, Refs. [60, 61] discuss how the different contributions to the spin structure of protons appear in the general parameterization

		Quark polarization		
		U	L	T
Nucleon polarization	U	f_1		$h_1^\perp$
	L		$g_{1\mathrm{L}}$	$h_{1\mathrm{L}}^\perp$
	T	$f_{1\mathrm{T}}^\perp$	$g_{1\mathrm{T}}$	$h_1, h_1^\perp$

Table 1.1: Quark TMD PDFs at leading twist: there are eight possible combinations to describe the polarization of the nucleon and the polarization of the quarks. The letters U, L, and T stand for unpolarized, longitudinally polarized and transversely polarized nucleons (rows) and corresponding quarks (columns).

of the correlator, as well as the explicit expression for the TMD PDF correlator.

At leading order (LO), the lowest order in the expansion in α_S of the terms that appear in the explicit expression of TMDs (see Ch. 2), TMDs have a probabilistic interpretation: for example, $f_1(x, k_\perp^2)$ represents the probability density to find an unpolarized quark of transverse momentum $k_\perp$ and longitudinal momentum fraction x in an unpolarized hadron.

Looking at Tab. 1.1, in addition to $f_1(x, k_\perp^2)$, other two diagonal elements are the helicity distribution $g_{1\mathrm{L}}(x, k_\perp^2)$, which describes longitudinally polarized quarks in a longitudinally polarized hadron, and the so–called transversity distribution, $h_1(x, k_\perp^2)$. These distributions do not vanish when integrated in the transverse quark momentum $k_\perp$: they have a collinear counterpart and this favored the study of $f_1(x, k_\perp^2)$ and $g_{1\mathrm{L}}(x, k_\perp^2)$. At the present time, little is known about the $k_\perp$ dependence of $g_{1\mathrm{L}}(x, k_\perp^2)$, and the transversity distribution $h_1(x, k_\perp^2)$ is still largely unknown.

Both the transversity $h_1(x, k_\perp^2)$ and the pretzelosity function $h_1^\perp(x, k_\perp^2)$ give their contribution for transversely polarized quark in a transversely polarized hadron. Transversity was first introduced in Ref. [62], but, although fundamental for the nucleon description, $h_{1\mathrm{T}}$ has long remained unmeasured because of its chirally odd nature that prevents its measurement in inclusive DIS. [2] Pretzelosity made its first appearance in Ref. [56], even tough the name pretzelosity was proposed later [64].

Off–diagonal elements of Tab. 1.1 are "genuine" transverse–momentum–dependent functions and vanish after $k_\perp$ integration.

Particularly intriguing are the the correlations between quark transverse momentum and nucleon transverse spin, and between quark transverse spin and its transverse momentum in an unpolarized nucleon, which are encoded in the Sivers TMD $f_{1\mathrm{T}}^\perp(x, k_\perp^2)$

[2] The transversity distribution can only be measured in conjunction with another chiral–odd object. One possibility is represented by Semi–Inclusive Deep Inelastic Scattering reactions, $(lN \to lHX)$, where at least one final state hadron is detected in coincidence with the scattered lepton. In Ref. [63] a novel way of accessing the transversity PDF is proposed.

		Quark polarization		
		U	L	T
Hadron polarization	U	D_1		$H_1^\perp$
	L		G_{1L}	$H_{1L}^\perp$
	T	$D_{1T}^\perp$	G_{1T}	$H_{1T}, H_{1T}^\perp$

Table 1.2: Quark TMD FFs at leading twist: there are eight possible combinations to describe the polarization of the outgoing hadron and the polarization of the fragmenting quarks. The letters U, L, and T stand for unpolarized, longitudinally polarized and transversely polarized hadrons (rows) and corresponding quarks (columns).

(introduced in Ref. [65]) and Boer–Mulders TMD $h_1^\perp(x, k_\perp^2)$ (Ref. [57]). These distributions are T–odd, meaning that they violate time–reversal invariance when omitting the presence of Wilson lines; functions that do not have this behavior are called T–even. Given the T–odd character of both Sivers and Boer–Mulders functions, these TMDs exhibit a peculiar process dependence: their sign is expected to be reversed when observed in Semi–Inclusive Deep Inelastic Scattering ($lN \to lHX$) or from Drell–Yan ($NN \to l^+l^-X$).

Finally, the worm–gear functions $g_{1T}(x, k_\perp^2)$ and $h_{1L}^\perp(x, k_\perp^2)$, introduced in Ref. [62], describe longitudinally polarized quarks in a transversely polarized hadron and transversely polarized quark in a longitudinally polarized hadron respectively. "Worm–gear" functions is an informal name, given to $g_{1T}(x, k_\perp^2)$ and $h_{1L}^\perp(x, k_\perp^2)$ in Ref. [56]. In Ref. [66] a way to access the functions $g_{1T}(x, k_\perp^2)$ and $h_{1L}^\perp(x, k_\perp^2)$ was emphasized.

Tab. 1.2 is analogous to Tab. 1.1 and contains TMD Fragmentation Functions (TMD FFs), which describe the various possible combinations for the polarization of the fragmenting[3] quark with respect to the polarization of the hadron in the final state. When the polarization in the final state is not accounted for, only two TMD FFs come into play: the unpolarized $D_1(z, P_\perp)$ and the Collins $H_1^\perp(z, P_\perp)$ fragmentation function, where z is the energy fraction carried by the final state hadron and $P_\perp$ is the transverse momentum of the observed produced hadron with respect to the the fragmenting parton.

In this work, we will focus on the unpolarized TMDs $f_1(x, k_\perp^2)$ and $D_1(z, P_\perp)$. Information on the functional form of TMDs can be obtained from Drell–Yan (DY), Semi–Inclusive Deep Inelastic Scattering (SIDIS) and e^+e^- annihilation. In fact, factorization theorems allow us to write the cross section for these processes in term of convolutions of TMDs. In particular, the DY cross section is proportional to a convolution of two TMD PDFs, the cross section of e^+e^- annihilation contains a convolution of

[3]The term "fragmentation" is used to denote a parton fragmenting into an observed hadron.

two TMD FFs and the SIDIS cross section can be expressed in terms of a convolution of one TMD PDF and one TMD FF.

Moreover, TMDs contribute directly to measurable azimuthal asymmetries.

TMDs are partially computable by means of well–established perturbative methods that take into account soft and collinear radiation to all orders. However, calculations based on perturbative QCD become unreliable for values of transverse momentum close to the Landau pole ($\Lambda_{\rm QCD}$). In this regime, nonperturbative components have to be included and have to be determined through fits to experimental data.

Several works in the past have studied the nonperturbative components in Drell–Yan [67–74] or in SIDIS [75, 76], without directly mentioning TMDs. More recent works directly performed extractions of TMDs from Drell-Yan data [77–79], semi-inclusive DIS data [80, 81] or both [82–85]. Alternatively, TMDs were determined in the so–called parton–branching approach by solving evolution equations with an iterative method similar to parton showers but including transverse momentum dependence [86, 87].

A precise knowledge of TMDs is useful not only to investigate the structure of the nucleon in greater detail, but also to improve the reliability of predictions involving TMDs. At high energies, the perturbative part of TMDs may be dominant, but when extreme precision is required, also the nonperturbative components become relevant (see, *e.g.*, Ref. [88]). To give an example of the need to produce precise predictions, we observed that the inclusion in TMD fits of the recent LHC data, from the ATLAS collaboration, already call for the maximum accuracy that theory can accomplish, as will be discussed in this thesis.

Among the challenges of present and future collider physics there certainly is the necessity to have computational tools able to reproduce experimental observables with an ever increasing perturbative accuracy. This means that the optimization of the computation of multidimensional integrals as well as the inclusion of more perturbative terms are a fundamental part of the efforts of hadron phenomenology. Presently, there are several tools and codes that can provide predictions and cross section computations at different levels of accuracy. In general, numerical frameworks for the computation of observables can differ for several aspects: for example, for the choices made for Landau pole regularization, for intrinsic k_T generation, for the approaches to transverse momentum (q_T) resummation, for the matching to collinear PDFs at large momentum, where TMD factorization is not expected to hold and collinear factorization is the appropriate framework.

`RadISH` [89, 90] provides predictions with an accuracy of N^{3}LL and matching to NNLO (the presence of the matching allows us to extend predictions to large transverse momentum). `ArTeMiDe` performs TMD fits, *i.e*, it computes predictions and calculates the χ^2 with respect to Drell–Yan and SIDIS data, using the so–called ζ–prescription,

with different accuracies (and up to what we defined, in Sec. 2.1.3, as NNLL′) [78, 85].

Other computational tools available are: `DYRes/DYTURBO`[74], a q_T resummation numerical program for the calculation of Drell—Yan collinear cross sections; `Cute` [73, 91], a C++ program to calculate the differential cross sections for electroweak gauge boson and Higgs boson production at small transverse momentum, implementing N^{3}LL resummation as well as matching to NNLO fixed–order results ; `ReSolve`[92], a Monte Carlo differential cross section and parton level event generator.

`NangaParbat`, the platform described in this thesis, can provide predictions up to N^{3}LL level and perform fits of Drell–Yan data using more standard prescriptions compared to `ArTeMiDe`.

Looking at the future of hadronic physics, one very important opportunity to explore the internal structure of the nucleon in three dimensions will be offered by the Electron—Ion Collider (EIC), a new facility which is about to be build at Brookhaven National Laboratory [93].

What will make the EIC a unique accelerator is the fact that it will be colliding polarized electrons off polarized protons and light nuclei, yielding the spin degrees of freedom necessary to pursue an ambitious physics program driven by the exploration of the spin structure and the acquisition of new information on multi–dimensional tomographic images of protons and nuclei [94].

In the future EIC era, the possibility to produce precise theoretical predictions (both from the perturbative and the nonperturbative side) is going to be crucial in the study of future experimental data, especially considering the foreseen luminosity of the new accelerator, coupled with its energy variability and reach.

1.2 Content of this thesis

The main topics of this thesis are: the description of the suite of computational tools that we developed for TMD studies (`NangaParbat`) and the extraction of TMD PDFs performed using NangaParbat and dubbed PV19 [95].

The author of this thesis was one of the main developers of the `NangaParbat` code and one of the main authors of the PV19 TMD extraction.

The dissertation is organized as follows. In Ch. 2, the TMD theoretical framework is illustrated, giving an introduction on the Drell–Yan process and Semi–Inclusive DIS (SIDIS), two of the main sources of information on TMDs. For both processes we discuss the structure of the observable, TMD evolution and their matching, resummation of large logarithms, the convention we used for the logarithmic ordering and the choices of the nonperturbative terms. In particular, both the PV19 and the PV17 [84] nonperturbative functional forms are discussed.

The structure of `NangaParbat`, the C++ fitting framework that we created, is illustrated in Ch. 3. At the moment of the writing of this thesis, `NangaParbat` has two main purposes: performing extractions of unpolarized TMD PDF form Drell–Yan data and generating TMD grids. The author of this thesis contributed to develop both these aspects of `NangaParbat`, also creating an user–oriented `Python` interface of the code.

One of the central characteristics of `NangaParbat` is the faithful treatment of experimental errors: the χ^2 is computed taking into account the nature (correlated or uncorrelated) of the uncertainties. Besides the implementation of the observable (the Drell–Yan cross section) and the computation of the perturbative terms, we also illustrate the optimized integration techniques we used in Ref. [95]. Finally, the last part of Ch. 3 is dedicated to the production and the interpolation of TMD and structure function grids, based on PV17 [84] and PV19 [95] extractions. The author of this thesis was responsible for the production and testing of the TMD PDF, TMD FF and unpolarized structure function grids discussed in this work. These tools are important to make predictions for observables that will be measured at the future Electron–Ion Collider (EIC) and have been used for the preparation of the EIC Yellow Report [96].

In Ch. 4, we discuss the results of the PV19 TMD extraction: we determined the unpolarized quark TMDs by fitting Drell–Yan data from experiments at Tevatron, RHIC, LHC, and low–energy experiments at Fermilab, for a total of around 350 data points. The dataset is similar to the one studied in Ref. [79], but there are some important differences: whenever available, we use cross section measurements without any normalization factor; TMD evolution is implemented in a different way; for the first time, TMD evolution is implemented up to next–to–next–to–next–to–leading logarithmic ($\mathrm{N^3LL}$) accuracy. Compared to Ref. [84], we exclude data from SIDIS, but we greatly extend the Drell–Yan dataset, we improve the logarithmic accuracy, we study normalizations with much greater care, and we abandon the narrow–width approximation for Z–boson production data. In conclusion, PV19 is one of the two state–of–the–art extractions of TMDs available at present.

Finally, Ch. 5 contains a brief overview of this thesis results.

Chapter 2

Theoretical framework

To describe processes that are sensitive to intrinsic parton transverse momentum, TMD factorization is needed. The theoretical framework of TMD factorization is more complicated than that of collinear factorization: in the calculations that lead to collinear factorization there are important cancellations that occur after integrations over the parton momentum are carried out. In a TMD context, integrals over parton transverse momentum are not performed and contributions that would cancel must be accounted for. Collins, Soper and Sterman (CSS) devised a useful formalism [97–99] for TMD factorization that deals with the main complications of transverse momentum dependence and provides a systematic treatment of pQCD over the full transverse momentum spectrum.

The Drell–Yan (DY) process, Semi–Inclusive Deep Inelastic Scattering (SIDIS) and back–to–back hadron production in electron–positron annihilation at small transverse momentum are classic examples of where CSS TMD factorization formalism can be applied and information on the functional form of TMDs can be obtained.

For a more complete overview of TMD definitions and TMD factorization we refer the reader to Refs. [67, 76, 100–102].

This chapter is dedicated to an introduction of the theoretical framework of Drell–Yan and SIDIS processes. The analysis of these processes is here illustrated in the CSS TMD factorization framework, as factorization theorems allow us to write the cross section of these processes in terms of convolutions of TMDs.

2.1 The Drell–Yan process

In this section we describe the theoretical framework of our Drell–Yan extraction in Ref. [95], with a brief overview of TMD factorization for the Drell–Yan process, the evolution of TMDs and their matching onto the collinear PDFs. The perturbative or-

dering we use in this thesis is also discussed, along with the perturbative elements of the factorized formula within the particular choice of the evolution scales embraced in this analysis. We explain the need for the introduction of a nonperturbative contribution that needs to be determined from data, and we discuss its particular functional form.

2.1.1 Drell–Yan cross section in TMD factorization

In an inclusive Drell–Yan process,

$$h_1(P_1) + h_2(P_2) \longrightarrow \gamma^*/Z(q) + X \longrightarrow \ell^+(l) + \ell^-(l') + X \ , \tag{2.1}$$

two hadrons, denoted in Fig. 2.1 as h_1 and h_2, with 4–momenta P_1 and P_2, collide with center–of–mass energy squared $s = (P_1 + P_2)^2$ and produce a neutral vector boson γ^*/Z with 4–momentum q and large invariant mass $Q = \sqrt{q^2}$.[1] The vector boson eventually decays into a lepton and antilepton pair, whose 4–momenta are constrained by momentum conservation, $q = l + l'$.

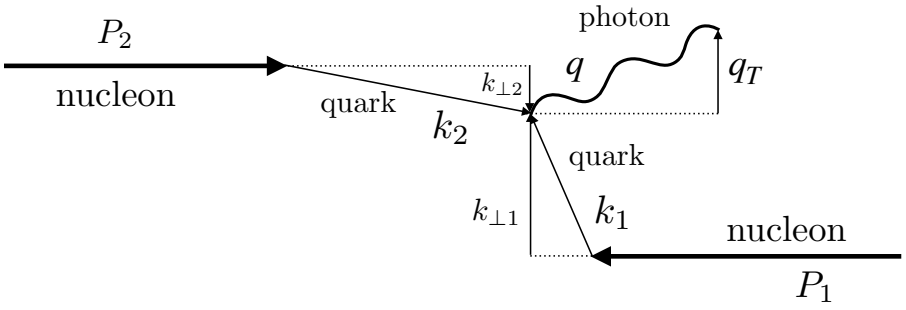


Figure 2.1: Relevant momenta involved in a Drell–Yan event. In a reference frame in which two colliding nucleons move along the z direction with 4–momenta P_1 and P_2, a quark with 4–momentum k_1 and transverse momentum $\boldsymbol{k}_{\perp 1}$ annihilates with a parton with 4–momentum k_2 and transverse momentum $\boldsymbol{k}_{\perp 2}$. A (virtual) photon (or Z) is produced with 4–momentum q and transverse momentum $\boldsymbol{q}_T = \boldsymbol{k}_{\perp 1} + \boldsymbol{k}_{\perp 2}$.

The absolute value of the transverse momentum and the rapidity of the Z boson (or, equivalently, of the lepton pair) are defined as

$$q_T = \sqrt{q_x^2 + q_y^2} \ , \qquad y = \frac{1}{2} \ln \left(\frac{q_0 + q_z}{q_0 - q_z} \right) \ , \tag{2.2}$$

[1] In general, γ^*/Z production is always associated also with the production of other resonances that can decay into a lepton–antilepton pair.

where the z direction is defined by the axis of the hadronic collision (see Fig. 2.1).

In this analysis, the object of interest is the transverse momentum distribution of the vector boson in the small–q_T region ($q_T \ll Q$). In this regime, the unpolarized differential cross section factorizes and can be expressed in terms of the unpolarized TMDs of the two hadrons as

$$\frac{d\sigma}{dQdydq_T} = \frac{16\pi^2\alpha_{\rm em}^2 q_T \mathcal{P}}{9Q^3} H(Q,\mu) \sum_q c_q(Q) \\ \times \int d^2\boldsymbol{k}_{\perp 1}\, d^2\boldsymbol{k}_{\perp 2}\, x_1 f_1^q\big(x_1, \boldsymbol{k}_{\perp 1}^2; \mu, \zeta_1\big)\, x_2 f_1^{\bar{q}}\big(x_2, \boldsymbol{k}_{\perp 2}^2; \mu, \zeta_2\big)\, \delta^{(2)}\big(\boldsymbol{k}_{\perp 1} + \boldsymbol{k}_{\perp 2} - \boldsymbol{q}_T\big)\ , \tag{2.3}$$

where $\alpha_{\rm em}$ is the electromagnetic coupling and $\mathcal{P}$ is the phase–space reduction factor that accounts for possible kinematic cuts on the final–state leptons (see Appendix 3.A).

The hard factor H represents the perturbative part of the hard scattering and depends on the hard scale Q and on the renormalization scale μ. The summation over q in Eq. (2.3) runs over the active quarks and antiquarks at the scale Q, and c_q are the respective electroweak charges given by

$$c_q(Q) = e_q^2 - 2e_q V_q V_\ell\, \chi_1(Q) + (V_\ell^2 + A_\ell^2)\,(V_q^2 + A_q^2)\, \chi_2(Q)\ , \tag{2.4}$$

with

$$\chi_1(Q) = \frac{1}{4\sin^2\theta_W \cos^2\theta_W} \frac{Q^2(Q^2 - M_Z^2)}{(Q^2 - M_Z^2)^2 + M_Z^2\Gamma_Z^2}\ , \tag{2.5}$$

$$\chi_2(Q) = \frac{1}{16\sin^4\theta_W \cos^4\theta_W} \frac{Q^4}{(Q^2 - M_Z^2)^2 + M_Z^2\Gamma_Z^2}\ , \tag{2.6}$$

where e_q, V_q, and A_q are respectively the electric, vector, and axial charges of the flavor q; V_ℓ and A_ℓ are the vector and axial charges of the lepton ℓ; θ_W is the weak mixing angle; M_Z and Γ_Z are mass and width of the Z boson.

The second line of Eq. (2.3) displays the convolution of the TMDs f_1^q and $f_1^{\bar{q}}$ of the hadrons h_1 and h_2, respectively. It describes the annihilation of a quark q, with longitudinal momentum fraction $x_1 = Qe^y/\sqrt{s}$ and transverse momentum $\boldsymbol{k}_{\perp 1}$, with the corresponding antiquark $\bar{q}$, with longitudinal momentum fraction $x_2 = Qe^{-y}/\sqrt{s}$ and transverse momentum $\boldsymbol{k}_{\perp 2}$. In the annihilation, the momentum conservation is guaranteed by the presence of $\delta^{(2)}\big(\boldsymbol{k}_{\perp 1} + \boldsymbol{k}_{\perp 2} - \boldsymbol{q}_T\big)$ (see Fig. 2.1).

As a consequence of renormalization and of the removal of the rapidity divergences [44], TMDs acquire a dependence on the renormalization scale μ and on the so–called rapidity scale ζ. Our choice for these scales is discussed in Sec. 2.1.3. Here,

we just remark that the rapidity scales ζ_1 and ζ_2 in Eq. (2.3) must obey the kinematic constraint $\zeta_1\zeta_2 = Q^4$.

It is convenient to rewrite the convolution in the conjugate position space by using the Fourier transform of each TMD, defined as[2]

$$\tilde{f}_1^q\big(x, b_T; \mu, \zeta\big) = \int d^2\boldsymbol{k}_\perp \, e^{i\boldsymbol{k}_\perp\cdot\boldsymbol{b}_T} \, f_1^q\big(x, \boldsymbol{k}_\perp^2; \mu, \zeta\big) \,, \tag{2.7}$$

where b_T is the absolute value of the vector $\mathbf{b}_T$ ($b_T = |\mathbf{b}_T|$). By using Eq. (2.7), we can rewrite the convolution of TMDs as

$$\begin{aligned}
&\int d^2\boldsymbol{k}_{\perp 1}\, d^2\boldsymbol{k}_{\perp 2}\, x_1 f_1^q\big(x_1, \boldsymbol{k}_{\perp 1}^2; \mu, \zeta_1\big)\, x_2 f_1^{\bar{q}}\big(x_2, \boldsymbol{k}_{\perp 2}^2; \mu, \zeta_2\big)\, \delta^{(2)}\big(\boldsymbol{k}_{\perp 1} + \boldsymbol{k}_{\perp 2} - \boldsymbol{q}_T\big) \\
&= \int \frac{d^2\boldsymbol{b}_T}{(2\pi)^2}\, e^{i\boldsymbol{b}_T\cdot\boldsymbol{q}_T}\, x_1 \tilde{f}_1^q\big(x_1, b_T; \mu, \zeta_1\big)\, x_2 \tilde{f}_1^{\bar{q}}\big(x_2, b_T; \mu, \zeta_2\big) \\
&= \frac{1}{2\pi} \int_0^\infty db_T\, b_T\, J_0\big(b_T q_T\big)\, x_1 \tilde{f}_1^q\big(x_1, b_T; \mu, \zeta_1\big)\, x_2 \tilde{f}_1^{\bar{q}}\big(x_2, b_T; \mu, \zeta_2\big) \,,
\end{aligned} \tag{2.8}$$

where J_0 is the 0–th order Bessel function of the first kind that has the following integral representation

$$J_0(x) = \frac{1}{2\pi} \int_0^{2\pi} d\theta \, e^{ix\cos\theta} \,. \tag{2.9}$$

By inserting Eq. (2.8) into the cross section in Eq. (2.3), we finally get

$$\begin{aligned}
\frac{d\sigma}{dQdyd\boldsymbol{q}_T} &= \frac{8\pi\alpha_{\rm em}^2 q_T \mathcal{P}}{9Q^3} H(Q,\mu) \\
&\times \sum_q c_q(Q) \int_0^\infty db_T\, b_T\, J_0\big(b_T q_T\big)\, x_1 \tilde{f}_1^q\big(x_1, b_T; \mu, \zeta_1\big)\, x_2 \tilde{f}_1^{\bar{q}}\big(x_2, b_T; \mu, \zeta_2\big) \,,
\end{aligned} \tag{2.10}$$

which is the formula actually implemented in our analysis of Drell–Yan data. More details about how we encoded the observable in `NangaParbat` are discussed in Ch. 3, Sec. 3.4.

[2]For simplicity, henceforth we will refer to the b_T–dependent function $\tilde{f}_1$ as to TMD but understanding that this is in fact the Fourier transform of the actual TMD f_1. Note that in Ref. [84] the variable ξ_T was used in place of b_T. The reason was to avoid confusion with the impact parameter used in the GPD literature for which the symbol b_T is typically used. In this thesis, we decided to use b_T as it is more common in the TMD, q_T–resummation, and SCET literature, keeping in mind that this is *not* the impact parameter but the Fourier conjugate variable of q_T. Finally, we notice that in Ref. [84] the Fourier transform was defined with an extra $1/(2\pi)$ factor.

2.1.2 TMD evolution and matching

In Eq. (2.10), the dependence of the TMDs $\tilde{f}_1^{q(\bar{q})}$ on the scales μ and ζ arises from the removal of the ultraviolet and rapidity divergences in their operator definition. Each dependence is controlled by an evolution equation:

$$\frac{\partial \ln \tilde{f}_1}{\partial \ln \mu} = \gamma(\mu, \zeta) \,, \qquad \frac{\partial \ln \tilde{f}_1}{\partial \ln \sqrt{\zeta}} = K(\mu) \,, \tag{2.11}$$

where γ is the anomalous dimension of the Renormalization Group (RG) evolution in μ, and K is the anomalous dimension of the Collins–Soper evolution in $\sqrt{\zeta}$ [99]. Notice that, for brevity, we have dropped the flavor index q and $\bar{q}$. Moreover, since in this section we will only be concerned with the dependence of $\tilde{f}_1$ on the scales μ and ζ, we will also temporarily drop the dependence on x and b_T. In addition to the evolution equations in Eq. (2.11), the rapidity anomalous dimension K obeys its own RG equation:

$$\frac{\partial K}{\partial \ln \mu} = -\gamma_K\big(\alpha_s(\mu)\big) \,, \tag{2.12}$$

where γ_K is known as cusp anomalous dimension. Since the crossed double derivatives of $\tilde{f}_1$ must be equal, using Eqs. (2.11) and (2.12) we also get

$$\frac{\partial \gamma}{\partial \ln \sqrt{\zeta}} = -\gamma_K\big(\alpha_s(\mu)\big) \,. \tag{2.13}$$

Using the point $\zeta = \mu^2$ as a boundary condition, the solution of this differential equation is

$$\gamma(\mu, \zeta) = \gamma_F\big(\alpha_s(\mu)\big) - \gamma_K\big(\alpha_s(\mu)\big) \ln \frac{\sqrt{\zeta}}{\mu} \,, \tag{2.14}$$

where $\gamma_F(\alpha_s(\mu)) \equiv \gamma(\mu, \mu^2)$. If the TMD $\tilde{f}_1$ is known at some starting scales μ_0 and ζ_0, the solution of the evolution equations in Eq. (2.11) reads

$$\tilde{f}_1(\mu, \zeta) = R\big[(\mu, \zeta) \leftarrow (\mu_0, \zeta_0)\big] \tilde{f}_1(\mu_0, \zeta_0) \,, \tag{2.15}$$

where the so–called Sudakov form factor R accounts for the perturbative evolution of $\tilde{f}_1$ and it is defined as

$$R\big[(\mu, \zeta) \leftarrow (\mu_0, \zeta_0)\big] = \exp\left\{ K(\mu_0) \ln \frac{\sqrt{\zeta}}{\sqrt{\zeta_0}} + \int_{\mu_0}^{\mu} \frac{d\mu'}{\mu'} \left[\gamma_F(\alpha_s(\mu')) - \gamma_K(\alpha_s(\mu')) \ln \frac{\sqrt{\zeta}}{\mu'} \right] \right\} \,. \tag{2.16}$$

We note that Eq. (2.16) can be implemented in various ways [103–106]. In this work, we follow the standard approach described in [44]. Moreover, we calculate all ingredients

involved in Eq. (2.16) by adopting a fully numerical approach.

An important property of the TMD $\tilde{f}_1$ is that at small values of b_T it can be matched onto the *collinear* PDF. Reinstating for clarity the x and b_T dependence, we can write the initial scale TMD PDFs at small values of b_T as:

$$\tilde{f}_1^i(x, b_T; \mu_0, \zeta_0) = \sum_{j=g,q(\bar{q})} \int_x^1 \frac{dy}{y} C_{ij}(y; \mu_0, \zeta_0) f_j\left(\frac{x}{y}, \mu_0\right) , \tag{2.17}$$

where f_j are the collinear PDFs (including the gluon) and C_{ij} are the so–called matching functions that are perturbatively computable and are currently known up to N^{3}LO, *i.e.* $\mathcal{O}(\alpha_s^2)$ [107]. Then, introducing the notation

$$\tilde{f}_1(x, b_T; \mu_0, \zeta_0) = \int_x^1 \frac{dy}{y} C(y, b_T; \mu_0, \zeta_0) f_1\left(\frac{x}{y}; \mu_0\right) \equiv \big[C \otimes f_1\big](x, b_T; \mu_0, \zeta_0) \, . \tag{2.18}$$

where the Mellin convolution is denoted by the symbol $\otimes$, the sum over flavors is implied and the matching function C has to be regarded as a matrix in flavor space multiplying a column vector of collinear PDFs, the actual evolved TMD becomes

$$\tilde{f}_1(x, b_T; \mu, \zeta) = R\big[b_T; (\mu, \zeta) \leftarrow (\mu_0, \zeta_0)\big] \big[C \otimes f_1\big](x, b_T; \mu_0, \zeta_0) \, . \tag{2.19}$$

Matching and evolution are affected by nonperturbative effects that become relevant at large b_T. In order to account for such effects, one usually introduces a phenomenological function $f_{\rm NP}$. In the CSS approach [44], the b_T–space TMDs get a multiplicative correction that does not depend on the flavor. In addition, the perturbative content of the TMDs is smoothly damped away at large b_T by introducing the so–called b_* prescription. These aspects are discussed in Sec. 2.1.5.

2.1.3 Perturbative content

In order to use Eq. (2.19) to insert evolved TMDs in phenomenological applications, we need to define the values of both the initial and final pairs of scales, (μ_0, ζ_0) and (μ, ζ). It turns out that in the $\overline{\rm MS}$ renormalization scheme there exists a particular scale,

$$\mu_b(b_T) = \frac{2e^{-\gamma_E}}{b_T} \, , \tag{2.20}$$

with γ_E the Euler constant, such that the rapidity anomalous dimension K and the matching coefficient C computed at $\mu_0 = \sqrt{\zeta_0} = \mu_b$ admit a pure perturbative expansion free of explicit logarithms of the scales. Therefore, μ_b provides a natural choice for μ_0 and $\sqrt{\zeta_0}$.

The final renormalization scale μ must match the one used in the hard factor H in Eq. (2.10). Therefore, μ has to be of order Q for avoiding large logarithms in H: we choose $\mu = Q$. Any variation of μ with respect to this choice can be accounted for by expanding the solution of the RG equation for the strong coupling α_s. The rapidity scales ζ_1 and ζ_2 in Eq. (2.10) are bound to comply with $\zeta_1\zeta_2 = Q^4$. Therefore, the natural choice is $\zeta_1 = \zeta_2 = Q^2$. However, we stress that any choice that fulfills this constraint leads to the same cross section. In fact, from Eq. (2.16) it is evident that the evolution factors R entering the two TMDs in Eq. (2.10) combine in such a way that the result only depends on the product $\zeta_1\zeta_2$.

After choosing the scales, we discuss the perturbative ingredients that result from this particular choice. We first consider the hard function H. Up to two–loop accuracy, its perturbative expansion is

$$H(Q,Q) = 1 + \sum_{n=1}^{2} \left(\frac{\alpha_s(Q)}{4\pi}\right)^n H^{(n)} \,. \tag{2.21}$$

The coefficients $H^{(n)}$ can be read off from, *e.g.*, Ref. [90]. When going beyond $\mathcal{O}(\alpha_s^2)$, the hard function acquires a non–trivial flavor structure (see, *e.g.*, Ref. [108]). As a consequence, H should in principle be moved inside the flavor sum in Eq. (2.10). However, in the present analysis we do not consider corrections beyond $\mathcal{O}(\alpha_s^2)$ and Eq. (2.10) is appropriate.

Then, we consider the matching function C introduced in Eq. (2.18). By making the flavor and x dependencies explicit, the C have the following perturbative expansion

$$C_{ij}(x, b_T; \mu_b, \mu_b^2) = \delta_{ij}\delta(1-x) + \sum_{n=1}^{\infty} \left(\frac{\alpha_s(\mu_b)}{4\pi}\right)^n C_{ij}^{(n)}(x) \,. \tag{2.22}$$

The coefficient functions $C_{ij}^{(n)}$ up to $n = 2$ have been computed in Refs. [109, 110]. They have been reported also in Ref. [108], where the authors have verified the consistency of the results. The calculation of the $\mathcal{O}(\alpha_s^3)$ corrections to the quark matching functions appeared recently in Ref. [107].

As for the anomalous dimensions K, γ_F, and γ_K in the Sudakov form factor in

Eq. (2.16), their perturbative expansions read, respectively,

$$\begin{aligned} K(\mu_b) &= \sum_{n=0}^{\infty} \left(\frac{\alpha_s(\mu_b)}{4\pi}\right)^{n+1} K^{(n)} , \\ \gamma_F(\alpha_s(\mu)) &= \sum_{n=0}^{\infty} \left(\frac{\alpha_s(\mu)}{4\pi}\right)^{n+1} \gamma_F^{(n)} , \\ \gamma_K(\alpha_s(\mu)) &= \sum_{n=0}^{\infty} \left(\frac{\alpha_s(\mu)}{4\pi}\right)^{n+1} \gamma_K^{(n)} . \end{aligned} \tag{2.23}$$

The coefficients $K^{(n)}$ are listed up to $n = 3$ in Ref. [110] and up to $n = 2$ in Ref. [108]. They differ by a factor -2 due to a different definition of K. Also the coefficients $\gamma_F^{(n)}$ are given in Refs. [108, 110] up to $n = 2$, and they differ by a minus sign due to a different definition of the anomalous dimension. Finally, the coefficients $\gamma_K^{(n)}$ were originally computed in Ref. [111] and are also given in Refs. [108, 110] up to $n = 2$, where the differ by a factor 2. The coefficient $\gamma_K^{(3)}$ has been recently computed in Refs. [112–114].

2.1.4 Logarithmic ordering

In this section, we discuss how to combine in a consistent way the perturbative ingredients of Eqs. (2.21)-(2.23) for the computation of the cross section in Eq. (2.10) (see also Refs. [115, 116]).

As is well known, TMD factorization provides resummation of large logarithms of Q/q_T or, equivalently, of Q/μ_b. The resummation is implemented in the Sudakov form factor R in Eq. (2.16) whose perturbative expansion reads

$$R = 1 + \sum_{n=1}^{\infty} \left(\frac{\alpha_s(Q)}{4\pi}\right)^{n} \sum_{k=1}^{2n} L^k R^{(n,k)} , \tag{2.24}$$

with

$$L = \ln \frac{Q^2}{\mu_b^2} . \tag{2.25}$$

Because of the inner sum running up to $2n$, Eq. (2.24) exposes the double–logarithmic nature of the resummation. This structure can be traced back to the evolution equations in Eq. (2.11) that resum *two different* categories of logarithms. However, our particular choice of the scales ($\mu_0 = \sqrt{\zeta_0} = \mu_b$ and $\mu = \sqrt{\zeta} = Q$) makes the two categories to coincide, producing up to two logarithms for each power of α_s. Consequently,

Eq. (2.24) must include all powers of α_s if the scales are such that $\alpha_s L^2 \gtrsim 1$.

The expansion (2.24) can be rearranged to define a logarithmic ordering as

$$R = 1 + \sum_{k=0}^{\infty} R_{\mathrm{N}^k\mathrm{LL}} \,, \tag{2.26}$$

with

$$R_{\mathrm{N}^k\mathrm{LL}} = \sum_{n=1+[k/2]}^{\infty} \left(\frac{\alpha_s(Q)}{4\pi}\right)^n L^{2n-k} R^{(n,2n-k)} \,, \tag{2.27}$$

where $[k/2]$ is the integer part of $k/2$. According to this definition, the term $k = 0$ in Eq.(2.26) gives the leading–logarithmic (LL) approximation, the term $k = 1$ gives the next–to–leading–logarithmic (NLL) approximation, and so on. Multiplication of $R_{\mathrm{N}^k\mathrm{LL}}$ by a power p of α_s gives

$$\left(\frac{\alpha_s(Q)}{4\pi}\right)^p R_{\mathrm{N}^k\mathrm{LL}} = \sum_{m=1+[(k+2p)/2]}^{\infty} \left(\frac{\alpha_s(Q)}{4\pi}\right)^m L^{2m-(k+2p)} R^{(m-p,2m-(k+2p))} \sim R_{\mathrm{N}^{k+2p}\mathrm{LL}} \,, \tag{2.28}$$

where the symbol $\sim$ means that the left– and right–hand sides have the same logarithmic accuracy. This step is relevant because in the cross section the Sudakov form factor, Eq. (2.26), can be multiplied by some power of α_s originating from the hard factor H and/or the matching functions C.

Equation (2.28) states that, at the cross section level, the inclusion of an additional power of α_s in the perturbative expansion of H and/or C implies a contribution two orders higher with respect to the leading term in the logarithmic expansion. For example, at LL and NLL accuracy the functions H and C can be computed at $\mathcal{O}(1)$, at NNLL and N^{3}LL they need to include the $\mathcal{O}(\alpha_s)$ corrections, and so on. This logarithmic counting is illustrated in the left panel of Fig. 2.2: the diagonal bands represent the terms included in each $R_{\mathrm{N}^k\mathrm{LL}}$, with $\mathcal{H}^{(n)}$ the perturbative coefficients of either H or C or a combination of the two.

The counting discussed above generally applies to any process whose amplitude factorizes in the appropriate limit, such as DY in the $q_T \ll Q$ limit (TMD factorization). However, in the specific case of DY (*i.e.*, inclusive with respect to soft–collinear QCD radiation) also the phase space for the emission of n real particles in b_T space factorizes (see, *e.g.*, Ref. [117]). This feature, along with the factorization of the amplitude in the $q_T \ll Q$ limit, allows one to *exponentiate* soft–collinear emissions such that the

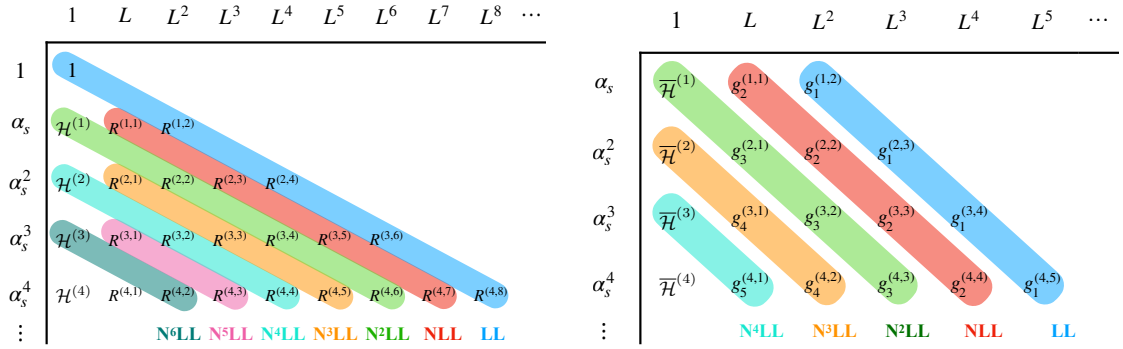


Figure 2.2: Graphical representation of logarithmic countings: in the left panel the counting is done at the level of the cross section, in the right panel at the level of the logarithm of the cross section.

Sudakov form factor can be written in the following general form (see, *e.g.*, Ref. [118])[3]

$$R = \exp\left[\frac{1}{2}Lg^{(1)}(\alpha_s L) + \frac{1}{2}g^{(2)}(\alpha_s L) + \frac{1}{2}\alpha_s g^{(3)}(\alpha_s L) + \dots\right] , \tag{2.29}$$

where the functions $g^{(i)}$ are such that $g^{(i)}(0) = 0$. As compared to the general counting in Eq. (2.24), exponentiation relates all the terms in Eq. (2.24) of the type $\alpha_s^n L^m$ with $n+1 < m \leq 2n$ to the lower–order terms. In Eq. (2.29), the logarithmic counting is performed at the level of the argument of the exponential. In this context, the terms $Lg^{(1)}$, $g^{(2)}$, $\alpha_s g^{(3)}$, etc., resum, respectively, the LL contributions $\alpha_s^n L^{n+1}$, the NLL contributions $\alpha_s^n L^n$, the NNLL contributions $\alpha_s^n L^{n-1}$, etc... Contrary to Eq. (2.24), this counting is driven by the condition $\alpha_s L \gtrsim 1$. This extends the validity of the calculation to smaller values of q_T relative to Q.

The logarithmic counting applied to the argument of the exponential is equivalent to consider the logarithm of the cross section [90]. In fact, neglecting for simplicity the matching functions, we schematically have

$$\ln\left(\frac{d\sigma}{dQdyd\boldsymbol{q}_T}\right) \propto \ln H + Lg^{(1)} + g^{(2)} + \alpha_s g^{(3)} + \dots \tag{2.30}$$

[3]The factors 1/2 in the argument of the exponential are justified by the fact that each of the two TMDs involved in the DY cross section contains an evolution factor R. In this way, Eq. (2.29) matches the literature on q_T–resummation where the Sudakov form factor is usually defined as the combination of both R's.

The logarithm of H can be expanded as

$$\ln(1+\alpha_s H^{(1)}+\alpha_s^2 H^{(2)}) = \alpha_s H^{(1)} + \alpha_s^2\left(H^{(2)} - \frac{H^{(1)2}}{2}\right) + \mathcal{O}(\alpha_s^3)\,. \tag{2.31}$$

The first term $\alpha_s H^{(1)}$ contributes to the tower $\alpha_s^n L^{n-1}$, that is the NNLL contribution. The second term $\alpha_s^2\left(H^{(2)} - H^{(1)2}/2\right)$ contributes to the $\alpha_s^n L^{n-2}$ tower, thus to the N^{3}LL contribution. The same counting applies to the matching functions C. The conclusion is that including $\mathcal{O}(\alpha_s)$ contributions in H and C implies introducing NNLL corrections, $\mathcal{O}(\alpha_s^2)$ contributions in H and C contribute to N^{3}LL accuracy, and so on. A graphical representation of this counting is sketched in the right panel of Fig. 2.2. Again, the bands represent the logarithmic towers, while $\overline{\mathcal{H}}^{(n)}$ are the appropriate coefficients of the expansion of either $\ln H$ or $\ln C$ or a combination. This logarithmic counting has been used in several papers (see, *e.g.*, Refs. [73, 90, 91, 119]). In this work, we will simply denote this counting with the acronyms NLL, NNLL, and so on, and for convenience we will refer to it as to "standard counting".

A slightly different counting has also been widely used in the literature (see, *e.g.*, Refs. [115, 120–123]). Expanding the Sudakov form factor (2.29) and multiplying it by the expansion of the hard function in Eq. (2.21), we obtain for the cross section

$$\frac{d\sigma}{dQdydq_T} \propto 1 + Lg^{(1)} + g^{(2)} + H^{(1)}\alpha_s L g^{(1)} + \dots\,, \tag{2.32}$$

where the rightmost term stems from the combination of the first–order terms $\alpha_s H^{(1)}$ and $Lg^{(1)}$ in both expansions. As it is clear from the previous discussion, this term has the same form $\alpha_s^n L^n$ as $g^{(2)}$. Then one can argue that NLL accuracy requires the inclusion not only of $g^{(2)}$ but also of $H^{(1)}$ [120]. This argument works to all orders: at any given logarithmic accuracy, it prescribes to include one more order in the perturbative expansion of H (and/or C) with respect to the standard counting. We will refer to this counting as the to "primed counting", denoting it as NLL′, NNLL′, and so on. The apparent contradiction between the standard and primed countings is resolved by observing that the first term of the perturbative expansion of $\alpha_s L g^{(1)}$ is proportional to $\alpha_s^2 L^2$. When considering the general expansion of the cross section given in Eqs. (2.26)–(2.28), a term proportional to $\alpha_s^2 L^2$ is of the form $\alpha_s^n L^{2n-2}$ and thus belongs to the NNLL tower. This is formally subleading with respect to the NLL accuracy determined by the $g^{(2)}$ term in the exponent.

Accurate predictions over a wide range in q_T require matching resummed calculations (valid at $q_T \ll Q$) to the corresponding fixed–order calculation (valid at $q_T \lesssim Q$). In this context, the primed ordering turns out to be more advantageous. Indeed, the accuracy of a fixed–order calculation is measured in terms of powers of α_s relative to

the leading term. In order to produce a Z boson with large q_T, it is necessary to produce (at least) a second object with large transverse momentum against which the Z boson recoils, *i.e.*, a jet. As a consequence, the leading–order (LO) contribution to the q_T distribution of the Z at fixed order is $\mathcal{O}(\alpha_s)$. The NLL′ prescription correctly reproduces the small–q_T limit of the LO fixed–order calculation. It is then possible to realize the matching in an *additive* way by combining the NLL′ resummed calculation with the LO fixed–order one (NLL′ + LO). The procedure can be extended to higher orders: NNLL′ + NLO, N^{3}LL′ + NNLO, and so on. Conversely, in the standard counting the matching to the LO fixed–order calculation requires to go further to NNLL accuracy (NNLL + LO), combining in this way a rather accurate calculation at small q_T with a poorly accurate calculation at large q_T. At higher orders one has N^{3}LL + NLO, N^{4}LL + NNLO, and so on. We remark that other forms of matching can be used to overcome the limitation of the standard counting [90, 124, 125].

Finally, Tab. 2.1 summarizes the perturbative ingredients to be used for a consistent computation of the cross section in Eq. (2.10) for both the standard and the primed countings. The numbers in Tab. 2.1 give the maximum power of α_s at which the corresponding quantity is to be computed, while the last column reports the corresponding accuracy in computing the evolution of the collinear PDFs and of the coupling α_s. In this analysis, we have used the PDF sets of the `MMHT2014` family [126] at the appropriate perturbative order accessed through the `LHAPDF` interface [127].

Accuracy	H and C	K and γ_F	γ_K	PDF and α_s evolution
LL	0	-	1	-
NLL	0	1	2	LO
NLL′	1	1	2	NLO
NNLL	1	2	3	NLO
NNLL′	2	2	3	NNLO
N^{3}LL	2	3	4	NNLO

Table 2.1: Truncation order in the expansions of Eqs. (2.21)-(2.23) for the two logarithmic countings considered in this work (see text). The last column reports the order used for the evolution of the collinear PDFs and α_s.

In the literature, in some cases, different notations to describe perturbative levels of accuracy are used. For example, in Refs. [120, 121, 128–130], the name NNLL refers to the logarithmic accuracy that in this work is called NNLL′ (given by the truncations in the second–to–last row of Tab. 2.1).

In Ref. [131], Scimemi and Vladimirov do not use the nomenclature of the first column of Tab. 2.1: in their TMD extraction they specify the perturbative orders used

for each part of the cross section, and, for example, they consider H at NNLL$'$ and γ_F at N^{3}LL.

2.1.5 Nonperturbative content and its parameterization

In the previous section, we noticed that in the $\overline{\text{MS}}$ scheme the rapidity evolution kernel K and the matching functions C can be made free of logarithms of the scales by introducing the natural scale μ_b defined in Eq. (2.20). Consistently, in the perturbative expansion of K (see first line of Eq. (2.23)) and C (see Eq. (2.22)) the strong coupling α_s must be computed at μ_b. For large values of b_T, μ_b becomes small such that $\alpha_s(\mu_b)$ may potentially become very large and eventually diverge when μ_b reaches the Landau pole at Λ_{QCD}. As a matter of fact, the integral in Eq. (2.10) does require accessing large values of b_T. It is then necessary to regularize this divergence by introducing a *prescription* that avoids integrating over the Landau pole. Different possibilities are available (see, *e.g.*, Refs. [125, 132]). In this thesis, we adopt the prescription originally proposed in Ref. [98]: we introduces the *arbitrary* parameter b_{max} that denotes the maximum value of b_T at which perturbation theory is considered reliable. Hence, b_{max} must be such that

$$\alpha_s\left(\frac{2e^{-\gamma_E}}{b_{\text{max}}}\right) \ll 1 \,. \tag{2.33}$$

Moreover, we also want to prevent μ_b from becoming much larger than the hard scale Q ($\mu_b \gg Q$). Despite not strictly mandatory (especially when considering only small values of q_T), this feature makes it possible to expand the cross section integrated in q_T, with the lowest–order term reproducing the lowest–order collinear result [133]. To this end, we define

$$b_{\text{min}} = \frac{2e^{-\gamma_E}}{Q} \,, \tag{2.34}$$

and

$$b_{\text{max}} = 2e^{-\gamma_E}\ \text{GeV}^{-1} \approx 1.123\ \text{GeV}^{-1}, \tag{2.35}$$

With the these choices, the scale μ_b is constrained between 1 GeV and Q, so that the collinear PDFs are never computed at a scale lower than 1 GeV and the lower limit of the integrals contained in the definition of the perturbative Sudakov factor (see Eq. (2.16)) can never become larger than the upper limit.

Then, we introduce a monotonic function $b_*(b_T)$ with the following asymptotic behaviors

$$\begin{aligned} b_*(b_T) &\to b_{\text{min}} \quad \text{for} \quad b_T \to 0 \,, \\ b_*(b_T) &\to b_{\text{max}} \quad \text{for} \quad b_T \to \infty \,. \end{aligned} \tag{2.36}$$

In this analysis, we adopt for $b_*(b_T)$ the same functional form chosen in Ref. [84] that

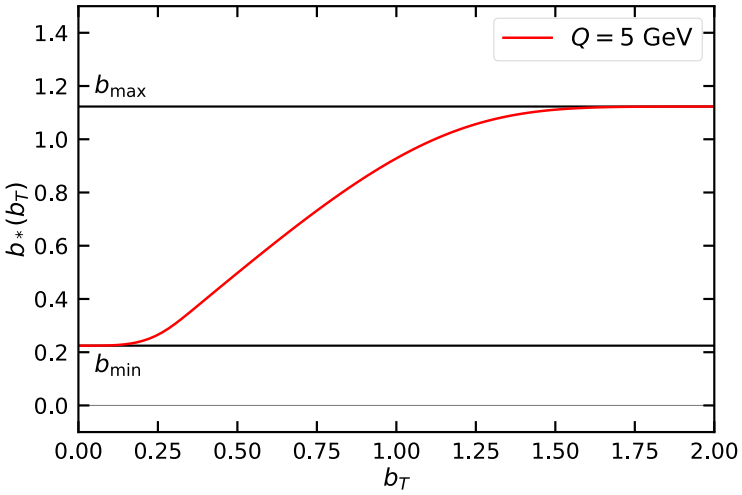


Figure 2.3: Example of the b_* prescription at $Q = 5$ GeV.

guarantees a smooth and rapid convergence towards the asymptotic limits:

$$b_*(b_T) = b_{\max}\left(\frac{1-\exp\left(-\frac{b_T^4}{b_{\max}^4}\right)}{1-\exp\left(-\frac{b_T^4}{b_{\min}^4}\right)}\right)^{\frac{1}{4}} . \tag{2.37}$$

There are other possible choices for the function $b_*(b_T)$: for example, Collins, Soper and Sterman in Ref. [98] used the following function, also discussed in [134–136]:

$$b_*(b_T) = \frac{b_T}{\sqrt{1+\frac{b_T^2}{b_{\max}^2}}} \tag{2.38}$$

with $b_{\max} \sim 0.5\,\mathrm{GeV}^{-1}$.

Now, we simply write the TMD $\tilde{f}_1$ as

$$\begin{aligned}\tilde{f}_1(x,b_T;\mu,\zeta) &= \left[\frac{\tilde{f}_1(x,b_T;\mu,\zeta)}{\tilde{f}_1(x,b_*(b_T);\mu,\zeta)}\right]\tilde{f}_1(x,b_*(b_T);\mu,\zeta) \\ &\equiv f_{\mathrm{NP}}(x,b_T,\zeta)\tilde{f}_1(x,b_*(b_T);\mu,\zeta)\,.\end{aligned} \tag{2.39}$$

This separation effectively defines f_{NP}. The advantage is that, due to the behavior of $b_*(b_T)$ for large values of b_T, $\tilde{f}_1(x,b_*(b_T),\mu,\zeta)$ remains in the perturbative region. The nonperturbative contributions are instead confined into f_{NP}, that has to be determined

through a fit to experimental data. However, using Eq. (2.39), we can work out some general properties of $f_{\rm NP}$. First, $f_{\rm NP}$ does not depend on the renormalization scale μ. To see this, using Eqs. (2.15) and (2.16) with $\mu_0 = \sqrt{\zeta_0} = \mu_b$, we find

$$\begin{aligned} f_{\rm NP}(x, b_T, \zeta) &= \frac{\tilde{f}_1(x, b_T; \mu, \zeta)}{\tilde{f}_1(x, b_*(b_T); \mu, \zeta)} = \exp\left\{ K(\mu_b) \ln \frac{\sqrt{\zeta}}{\mu_b} - K(\mu_{b_*}) \ln \frac{\sqrt{\zeta}}{\mu_{b_*}} \right. \\ &+ \left. \int_{\mu_b}^{\mu_{b_*}} \frac{d\mu'}{\mu'} \left[\gamma_F(\alpha_s(\mu')) - \gamma_K(\alpha_s(\mu')) \ln \frac{\sqrt{\zeta}}{\mu'} \right] \right\} \frac{\tilde{f}_1(x, b_T; \mu_b, \mu_b^2)}{\tilde{f}_1(x, b_*(b_T); \mu_{b_*}, \mu_{b_*}^2)} \,, \end{aligned} \tag{2.40}$$

with $\mu_{b_*} \equiv \mu_b(b_*(b_T))$. The dependence on μ evidently cancels in the ratio. In addition, for large values of b_T, μ_{b_*} saturates to some minimal value while μ_b becomes increasingly small. As a consequence of this departure between μ_{b_*} and μ_b, as well as between $\sqrt{\zeta}$ and μ_b, the exponential in Eq. (2.40) tends to be suppressed, and so does $f_{\rm NP}$. Conversely, as b_T becomes small b_* approaches $b_{\rm min}$. Using the definition in Eq. (2.34), it follows that μ_{b_*} saturates to Q while μ_b becomes larger and larger. In this limit, we have [133]

$$f_{\rm NP} \underset{b_T \to 0}{\longrightarrow} 1 + \mathcal{O}\left(\frac{1}{Q^p}\right) \,, \tag{2.41}$$

where p is some positive number. Since TMD factorization applies to leading–power in q_T/Q, we can neglect the power suppressed contribution such that $f_{\rm NP} \to 1$ for $b_T \to 0$. It is important to stress that the separation between perturbative and nonperturbative components of a TMD is *arbitrary* and depends on the particular choice of b_* (or in general on the prescription used to regularize the Landau pole). For any given choice, only the combination in Eq. (2.39) is meaningful, and it is misleading to refer to $f_{\rm NP}$ as to the nonperturbative part of TMDs in a universal sense. Following the requirements discussed above, we parameterize $f_{\rm NP}$ as

$$\begin{aligned} f_{\rm NP}(x, b_T, \zeta) &= \left[\frac{1-\lambda}{1 + g_1(x) \frac{b_T^2}{4}} + \lambda \exp\left(-g_{1B}(x) \frac{b_T^2}{4} \right) \right] \\ &\times \exp\left[-\left(g_2 + g_{2B} b_T^2\right) \ln\left(\frac{\zeta}{Q_0^2}\right) \frac{b_T^2}{4} \right] \,, \end{aligned} \tag{2.42}$$

with $Q_0 = 1$ GeV and with the $g_1(x)$ and $g_{1B}(x)$ functions given by

$$
\begin{aligned}
g_1(x) &= \frac{N_1}{x\sigma} \exp\left[-\frac{1}{2\sigma^2} \ln^2\left(\frac{x}{\alpha}\right)\right] , \\
g_{1B}(x) &= \frac{N_{1B}}{x\sigma_B} \exp\left[-\frac{1}{2\sigma_B^2} \ln^2\left(\frac{x}{\alpha_B}\right)\right] .
\end{aligned}
\tag{2.43}
$$

There are a total of 9 free parameters ($\lambda, g_2, g_{2B}, N_1, \sigma, \alpha, N_{1B}, \sigma_B, \alpha_B$) to be determined from data.

Apart from the logarithmic dependence on ζ, the functional form (2.42) is motivated by empirical considerations. The first line parameterizes the "intrinsic" TMD nonperturbative contribution and it only depends on x and b_T. The second line accounts for the nonperturbative correction to the perturbative evolution. Therefore, it only depends on b_T (on top of the known dependence on ζ).

The intrinsic contribution is a combination of a q-Gaussian (or Tsallis) distribution (first term) and a standard Gaussian distribution (second term). The q–Gaussian has a larger tail than the standard Gaussian, meaning that it gives a bigger contribution to the TMD at small transverse momentum. We found that this combination is able to reproduce the behavior at very small q_T of the experimental distributions from the lowest to the highest energies considered in our analysis.

The functions g_1 and g_{1B} in Eq. (2.43) are related to the width of the TMD distribution. Their are expected to depend on x on the basis of model calculations (see Ref. [137] and references therein) and more generally from Lorentz invariance constraints on the proton light–front wave functions (see, *e.g.*, the discussion in Ref. [138]). To best describe experimental data, we found it necessary to have wider TMDs at intermediate x. A log–normal dependence of g_1 and g_{1B} allowed us to properly describe the datasets differential in the boson rapidity y. In fact, as we will show below, the x dependence of $f_{\rm NP}$ is almost entirely determined by the ATLAS datasets, the only ones differential in y. Our present results are quite different from the ones obtained through fits to SIDIS data [84]. We expect that the addition of further datasets from DIS experiments [139, 140] will provide more sensitivity to the x dependence and possibly lead to different results.

The nonperturbative components of the TMDs could depend also on flavor [80, 88, 141]. However, in this work we refrain from including such dependence since DY data are not very sensitive to it. We stress that the fact that we can achieve a good description of data does not exclude the presence of a flavor dependence, which is actually expected on the basis of model calculations [142–147], lattice QCD studies [148], and also if QED corrections are taken into account [149, 150]. Higher sensitivity to flavor dependence may be provided again by SIDIS data with different targets and final–state hadrons and possibly by W–boson production data [151].

Concerning the b_T dependence of the nonperturbative evolution in the second line of Eq. (2.42), we have used a customary quadratic term [67, 71, 75, 152] with an additional quartic term. The latter contribution appears to be useful to reproduce the energy evolution displayed by the data. Other choices of the functional form have been discussed in, *e.g.*, Refs. [83, 134, 153, 154]. This contribution could be also determined using lattice QCD [155].

2.2 Semi–Inclusive Deep Inelastic Scattering

In recent years, single–inclusive deep–inelastic scattering lepton–nucleon scattering has emerged as an important tool to probe several aspects of perturbative and nonperturbative QCD. In one–particle semi–inclusive DIS (SIDIS), a lepton ℓ with momentum l scatters off a hadron target N with momentum P and mass M (the lepton mass can be neglected). The process is:

$$\ell(l) + N(P) \to \ell(l') + h(P_h) + X \tag{2.44}$$

where h is the detected hadron and X is a standard notation that indicates other products of the process that are not detected. The four–momenta associated with each particle are indicated in parentheses. In experiments, N is either a proton or a heavier nucleus. The data we use for nucleon's studies come from light nuclei, like deuterium and helium.[4]

It is convenient to define the momentum transfer $q = l - l'$ (the momentum of the exchanged photon), the invariant momentum transfer Q as

$$Q^2 = -q^2 \geq 0 \tag{2.45}$$

and the following combinations of variables:

$$x = \frac{Q^2}{2\,P\cdot q}, \qquad y = \frac{P\cdot q}{P\cdot l}, \qquad z = \frac{P\cdot P_h}{P\cdot q}, \qquad \gamma = \frac{2Mx}{Q}. \tag{2.46}$$

Here x is the Bjorken scaling variable, which corresponds to the fraction of the initial hadron momentum that is carried by the struck parton, y is the fraction of energy transferred from the lepton to the nucleon and therefore lies between 0 and 1, z is a variable that expresses the fractional longitudinal momentum of the detected hadron relative to its parent quark.

The diagram in Fig. 2.4 describes the relevant momenta involved in SIDIS (see also

[4] Scattering on a nucleus is often approximated as scattering on an isoscalar combination of protons and neutrons.

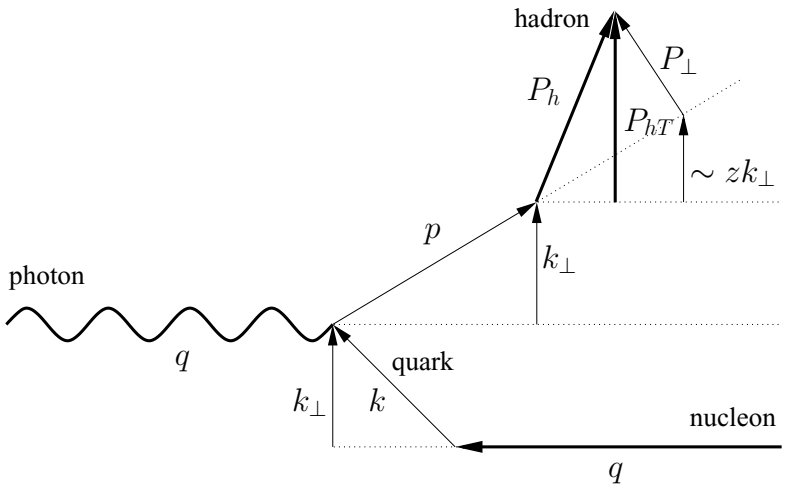


Figure 2.4: Illustration of the kinematics of SIDIS. The photon momentum defines the z axis (reference axis) and the struck quark in the nucleon has initial intrinsic transverse momentum $\boldsymbol{k}_\perp$ (not measured) with respect to the reference axis. The kinematics of the quark fragmentation process is depicted: the emitted hadron h carries a transverse momentum $\mathbf{P}_{hT}$ (with respect to the reference axis) which can be decomposed into two contributions: the quark transfers a fraction z of its transverse momentum $\mathbf{k}_\perp$ to the hadron and the hadron also acquires a further momentum $\mathbf{P}_\perp$ (not measured) transverse to the direction of the quark's momentum.

Ref. [156]). Both the longitudinal momentum fraction z and the transverse momentum P_{hT} of the hadron can be measured. Moreover, up to corrections of order $\mathcal{O}(k_\perp^2/Q^2)$, the following relation holds [157]:

$$\mathbf{P}_{hT} \approx z\boldsymbol{k}_\perp + \boldsymbol{P}_\perp \tag{2.47}$$

When the transverse momentum of the hadron is much smaller than the scale Q of the process ($\boldsymbol{P}_{hT}^2 \ll Q^2$) and in the kinematic limits $M^2 \ll Q^2$, the cross section for SIDIS can be expressed in a factorized form in terms of TMDs. As for the Drell–Yan case, the analysis of SIDIS in this work is done in the TMD factorization framework [44, 98]. A first application of TMD factorization to SIDIS was made in Ref. [75] and a comparison between this theory and data from HERA collider was made in Ref. [76].

The TMD formalism is in principle valid only when $|\boldsymbol{q}_T|^2 \ll Q^2$, where $\boldsymbol{q}_T$ is the transverse momentum of the exchanged photon in the frame where the target hadron and the observed hadron are collinear. The relation with the transverse momentum $\boldsymbol{P}_{hT}$ is $\boldsymbol{q}_T = -\boldsymbol{P}_{hT}/z$, so that the constraint becomes $P_{hT}^2 \ll z^2\,Q^2$. This constraint needs to be taken into account when calculating the structure functions, in particular when considering the z, P_{hT}, Q dependence.

2.2.1 Structure of the observable

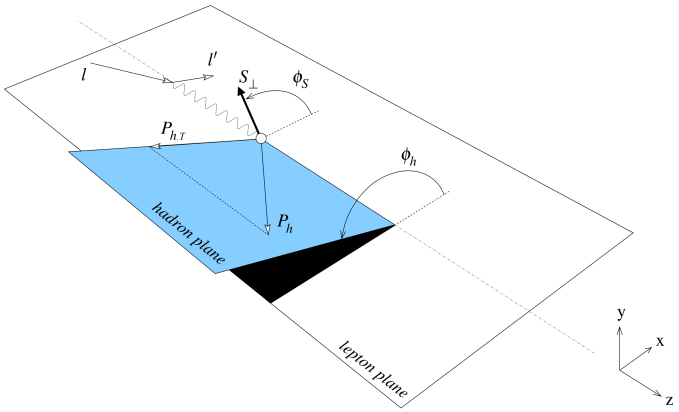


Figure 2.5: Definition of azimuthal angles for semi–inclusive deep inelastic scattering in the target rest frame: ϕ_h the azimuthal angle of $\mathbf{P}_{hT}$ relative to the lepton plane, while the angle ϕ_S is defined as the angle between the lepton scattering plane and the target–polarization vector $S_\perp$ of the transversely polarized nucleon.

The unpolarized cross section for SIDIS, integrated over the azimuthal angles ϕ_S and ϕ_h defined in Fig. 2.5, can be written as:

$$\frac{d\sigma}{dxdydzd^2P_{hT}} = \frac{4\pi^2\alpha^2}{2xQ^2}\frac{y}{(1-\varepsilon)}\Big[F_{UU,T}(x,z,P_{hT}^2,Q^2) + \varepsilon F_{UU,L}(x,z,P_{hT}^2,Q^2)\Big] \tag{2.48}$$

The ratio ε of longitudinal and transverse photon flux in Eq. (2.48) is defined as

$$\varepsilon = \frac{1-y-\frac{1}{4}\gamma^2y^2}{1-y+\frac{1}{2}y^2+\frac{1}{4}\gamma^2y^2}\,, \tag{2.49}$$

where y, γ are defined in Eq. (2.46). As for the notations in Eq. (2.48), $F_{XY,Z}$ indicates the structure function of an hadron target with polarization Y, a lepton beam with polarization X and a virtual photon exchanged in the polarization state Z.

In the kinematic limits $M^2 \ll Q^2$ and $\boldsymbol{P}_{hT}^2 \ll Q^2$, the structure function $F_{UU,L}$ of Eq. (2.48) can be neglected[5] [158].

To express the SIDIS structure functions in terms of TMD PDFs and FFs, we rely on factorization theorems [44, 98–101, 159–161] which at low transverse momenta allow us to write $F_{UU,T}$ as (see Fig. 2.4 for a graphical representation of the involved

[5]It is nonzero at twist 4

transverse momenta):

$$F_{UU,T}(x, z, \boldsymbol{P}_{hT}^2, Q^2) = \sum_q H_{UU,T}^q(Q, \mu) \tag{2.50}$$
$$\times\, x \int d^2\boldsymbol{k}_\perp\, d^2\boldsymbol{P}_\perp\, f_1^q\big(x, \boldsymbol{k}_\perp^2; \mu, \zeta_1\big)\, D_1^{q\to h}\big(z, \boldsymbol{P}_\perp^2; \mu, \zeta_2\big)\, \delta^{(2)}\big(z\boldsymbol{k}_\perp - \boldsymbol{P}_{hT} + \boldsymbol{P}_\perp\big)$$
$$+ Y_{UU,T}\big(Q^2, \boldsymbol{P}_{hT}^2\big) + \mathcal{O}\big(M^2/Q^2\big)\,.$$

Here $H_{UU,T}^q$ is the hard scattering part; $f_1^q(x_B, \mathbf{k}_\perp^2; \mu, \zeta_1)$ is the TMD PDF for an unpolarized parton of flavor q in an unpolarized proton, carrying longitudinal momentum fraction x and transverse momentum $\mathbf{k}_\perp$ at the factorization scale μ^2, which we choose to be equal to Q^2.

The function $D_1^{q\to h}(z, \mathbf{P}_\perp^2; \mu, \zeta_2)$ is the TMD fragmentation function for an unpolarized parton of flavor q that fragments into an unpolarized hadron h (detected) carrying longitudinal momentum fraction z and transverse momentum $\mathbf{P}_\perp$, at the factorization scale $\mu^2 = Q^2$.

As for the Drell–Yan case, we choose $\zeta_1 = \zeta_2 = Q^2$.

The term $Y_{UU,T}$ is introduced to ensure a matching to the perturbative calculations at high transverse momentum and accounts for the large transverse momentum dependence of the cross section, where the approximations needed for TMD factorization break down. There, collinear factorization becomes the appropriate framework.

Here we limit to the case $Q \ll M_Z$ such that we can neglect the contribution of the Z boson and thus the electroweak couplings are given by the squared electric charges.

Considering only the lowest order contributions (the terms at order α_s^0), Eq (2.50) can be written as

$$F_{UU,T}(x, z, \mathbf{P}_{hT}^2; Q^2) = \sum_q e^2 \left[f_1^q \otimes D_1^{q\to h}\right](x, z, \mathbf{P}_{hT}^2; Q^2) \tag{2.51}$$

where the convolution $\left[f_1^q \otimes D_1^{q\to h}\right]$ is:

$$\left[f_1^q \otimes D_1^{q\to h}\right](x, z, \mathbf{P}_{hT}^2; Q^2) =$$
$$= x \int d^2\boldsymbol{k}_\perp\, d^2\boldsymbol{P}_\perp\, f_1^q\big(x, \boldsymbol{k}_\perp^2; \mu, \zeta_1\big)\, D_1^{q\to h}\big(z, \boldsymbol{P}_\perp^2; \mu, \zeta_2\big)\, \delta^{(2)}\big(z\boldsymbol{k}_\perp - \boldsymbol{P}_{hT} + \boldsymbol{P}_\perp\big) \tag{2.52}$$

Elements of Eq. (2.50) can be expanded in terms of α_s in a way analogous to the one shown in Sec. 2.1.4. In particular, in this case, perturbative corrections include large logarithms $L \equiv \log\left(z^2Q^2/P_{hT}^2\right)$, so that $\alpha_S L \approx 1$. In the analysis in Ref.[84], only leading and next–to–leading logarithms are taken into account (NLL).

In these approximations (α_S^0 and NLL), only the first term in Eq. (2.50) is relevant (often in the literature this has been called W term) and we expect this term to provide a good description of the structure function in the region where $P_{hT}^2 \ll Q^2$. [6] In Ref. [84], the most recent extraction of TMDs form the Pavia group that takes into account SIDIS data, all corrections of order M^2/Q^2 have been neglected: this assumption is called *leading twist approximation*.

In order to apply TMD evolution equations, we calculate the Fourier transform of the part of Eq. (2.50) involving TMDs. The structure function thus reduces to

$$F_{UU,T}(x,z,P_{hT}^2;Q^2) = \frac{1}{2\pi}\, x \sum_q e_q^2 \times \int_0^{+\infty} db_T\, b_T\, J_0\left(\frac{b_T P_{hT}}{z}\right) \widetilde{f}_1^q(x,b_T^2;\mu,\zeta_1)\, \widetilde{D}_1^{q\to h}(z,b_T^2;\mu,\zeta_2)\,. \tag{2.53}$$

where we introduced the Fourier transforms of the TMD PDF and FF according to

$$\tilde{f}_1^q\big(x,b_T^2;\mu,\zeta_1\big) = \int_0^{\infty} d|\boldsymbol{k}_\perp||\boldsymbol{k}_\perp| J_0\big(b_T|\boldsymbol{k}_\perp|\big) f_1^a\big(x,\boldsymbol{k}_\perp^2;\mu,\zeta_1\big), \tag{2.54}$$

$$\tilde{D}_1^{q\to h}\big(z,b_T^2;\mu,\zeta_2\big) = \int_0^{\infty} \frac{d|\boldsymbol{P}_\perp|}{z^2} |\boldsymbol{P}_\perp| J_0\big(b_T|\boldsymbol{P}_\perp|/z\big) D_1^{q\to h}\big(z,\boldsymbol{P}_\perp^2;\mu,\zeta_2\big). \tag{2.55}$$

The Bessel function of order 0-th of the first kind J_0 is defined in Eq. (2.9).

As for the Drell–Yan case in Sec. 2.1.2, the single TMDs are evolved and matched onto the respective collinear functions:

$$\begin{aligned}\overline{F}_q(x,b;\mu,\zeta) &\equiv x f_1^q(x,b;\mu,\zeta) \\ &= R_q(\mu_0,\zeta_0\to\mu,\zeta;b) \sum_j \int_x^1 dy\, C_{qj}(y;\mu_0,\zeta_0) \left[\frac{x}{y} f_j\left(\frac{x}{y},\mu_0\right)\right],\end{aligned} \tag{2.56}$$

and

$$\begin{aligned}\overline{D}_q(z,b;\mu,\zeta) &\equiv z^3 D_1^q(z,b;\mu,\zeta) \\ &= R_q(\mu_0,\zeta_0\to\mu,\zeta;b) \sum_j \int_z^1 dy\, \left[y^2 C_{qj}(y;\mu_0,\zeta_0)\right] \left[\frac{z}{y} d_j\left(\frac{z}{y},\mu_0\right)\right].\end{aligned} \tag{2.57}$$

Low–momentum nonperturbative corrections are taken into account by introducing the

[6] We note that it can happen that $Y_{UU,T}$, defined in the standard way (see, e.g., Ref. [98]), gives large contributions also in this region, but it is admissible to redefine it in order to avoid this problem [133].

monotonic function $b_*(b_T)$, defined in Sec. 2.1.5, that behaves as:

$$\lim_{b\to 0} b_*(b_T) = b_{\min} \quad \text{and} \quad \lim_{b\to\infty} b_*(b_T) = b_{\max}\,. \tag{2.58}$$

This allows us to replace the TMDs in Eq. (2.53) with their "regularized" version:

$$\begin{aligned} \tilde{f}_1(x, b_T; \mu, \zeta) &\to \tilde{f}_1(x, b_*(b_T); \mu, \zeta) f_{\rm NP}(x, b_T, \zeta)\,, \\ \tilde{D}_1(z, b_T; \mu, \zeta) &\to \tilde{D}_1(z, b_*(b_T); \mu, \zeta) D_{\rm NP}(z, b_T, \zeta)\,, \end{aligned} \tag{2.59}$$

where we have introduced the nonperturbative functions $f_{\rm NP}$ and $D_{\rm NP}$. It is important to stress that these functions further factorize as follows:

$$\begin{aligned} f_{\rm NP}(x, b_T, \zeta) &= \hat{f}_{\rm NP}(x, b_T) \exp\left[g_K(b_T) \ln\left(\frac{\zeta}{Q_0^2}\right)\right]\,, \\ D_{\rm NP}(z, b_T, \zeta) &= \hat{D}_{\rm NP}(x, b_T) \exp\left[g_K(b_T) \ln\left(\frac{\zeta}{Q_0^2}\right)\right]\,. \end{aligned} \tag{2.60}$$

The common exponential function represents the nonperturbative corrections to TMD evolution, is flavor independent, x and z independent and is equal for TMD PDFs and TMD FFs. The specific functional form of the nonperturbative part of the evolution is driven by the solution of the Collins–Soper equation where Q_0 is some initial scale: the kernel can not be computed in the nonperturbative region, and hence has to be extracted from data. The set of nonperturbative functions to be determined from fits to data are $\hat{f}_{\rm NP}$, $\hat{D}_{\rm NP}$, and $g_K(b_T)$. It is worth noticing that by definition

$$D_{\rm NP}(z, b_T, \zeta) = \frac{\overline{D}_q(z, b_T; \mu, \zeta)}{\overline{D}_q(z, b_*(b_T); \mu, \zeta)}\,, \tag{2.61}$$

and similarly for $f_{\rm NP}$. Therefore, one has a partial handle on the b_T-dependence of these functions from the region in which b_T is small enough to make both numerator and denominator perturbatively computable.

Following Refs. [71, 72, 76], the nonperturbative Sudakov factor in PV17 (Ref. [84]) is

$$g_K(b_T) = -g_2 b_T^2/2 \tag{2.62}$$

with g_2 a free parameter. Several alternative forms have been proposed [134, 153]. Also, recent theoretical studies aimed at calculating this term using nonperturbative methods [162].

In PV17 analysis, for the collinear PDFs f_1^a the GJR08FFnloE set [163] was adopted, obtained through the LHAPDF library [127], and for the collinear fragmentation func-

tions the DSS14 NLO set for pions [164] and the DSS07 NLO set for kaons [165] were chosen.

SIDIS multiplicities

In order to extract $f_{\rm NP}$ and $D_{\rm NP}$, a fit to data is necessary. For semi–inclusive DIS the available data released from the COMPASS [166] and HERMES [139] collaborations refer to *hadron multiplicities*, namely to the differential number of hadrons produced per corresponding inclusive deep–inelastic scattering event, like In terms of cross sections, the multiplicities can be expressed as [80]:

$$m_N^h(x,z,\mathbf{P}_{h\perp}^2,Q^2) = \frac{d^6\sigma_N^h/dxdzd\mathbf{P}_{h\perp}^2dQ^2}{d^3\sigma_{DIS}/dxdQ^2} \tag{2.63}$$

where $d\sigma_N^h$ is the differential cross section for the semi–inclusive DIS process and $d\sigma_{DIS}$ is the corresponding inclusive one.

In the single photon exchange approximation, assuming a complete integration over the azimuthal angle ϕ_s of the detected hadron, multiplicities can be written as ratios of structure functions

$$m_N^h(x,z,\mathbf{P}_{h\perp}^2,Q^2) = \frac{\pi F_{UU,T}(x,z,\mathbf{P}_{h\perp}^2,Q^2) + \pi\varepsilon F_{UU,L}(x,z,\mathbf{P}_{h\perp}^2,Q^2)}{F_T(x,Q^2)+\varepsilon F_L(x,Q^2)} \tag{2.64}$$

In the denominator, only the photon polarization is explicitly written (T, L), as usually done in the literature. The structure function F_L in the denominator contains contributions involving powers of the strong coupling constant α_S at an order that goes beyond the level reached in the analysis in Ref. [84]; hence, it is neglected (for measurements and estimates of the F_L structure function see, e.g., Refs. [167, 168] and references therein).

2.2.2 Nonperturbative functions

We illustrate here the choices of Ref. [84] for the nonperturbative terms $f_{\rm NP}$ and $D_{\rm NP}$ of Eq. (2.60). The notation used here is slightly different from the one used in Ref. [84], in order to be consistent with the definition of the Fourier–Bessel transform in Eq. (2.9) and with the definitions of nonperturbative functions in Eq. (2.60). The function $f_{\rm NP}$ in PV17 is:

$$f_{\rm NP}^q(x,b_T^2) = e^{-g_2\frac{b_T^2}{2}\ln\left(\frac{\zeta}{Q_0^2}\right)} e^{-g_{1q}\frac{b_T^2}{4}}\left(1-\frac{\lambda g_{1q}^2}{1+\lambda g_{1q}}\frac{b_T^2}{4}\right), \tag{2.65}$$

while the nonperturbative part of the TMD FF is:

$$D_{\rm NP}^{q\to h}(z,b_T^2) = e^{-g_2\frac{b_T^2}{2}\ln\left(\frac{\zeta}{Q_0^2}\right)} \times \frac{g_{3q\to h}\; e^{-g_{3q\to h}\frac{b_T^2}{4z^2}} + \left(\lambda_F/z^2\right) g_{4q\to h}^2 \left(1 - g_{4q\to h}\frac{b_T^2}{4z^2}\right)\; e^{-g_{4q\to h}^2\frac{b_T^2}{4z^2}}}{z^2\left(g_{3q\to h} + \left(\lambda_F/z^2\right) g_{4q\to h}^2\right)} \,. \tag{2.66}$$

In the nonperturbative Sudakov factor, which is the same for both expressions, we made use of Eq. (2.62); the scale ζ is chosen equal to Q^2 and $Q_0 = 1$ GeV.

The dependence on the fractional longitudinal momentum x of the TMD Gaussian width of the TMD in PV17 is given by

$$g_1(x) = N_1 \frac{(1-x)^\alpha\; x^\sigma}{(1-\hat{x})^\alpha\; \hat{x}^\sigma} \,, \tag{2.67}$$

where α, σ, and N_1 ($\equiv g_1(\hat{x})$ with $\hat{x} = 0.1$), are free parameters. Similarly, for fragmentation functions we have

$$g_{3,4}(z) = N_{3,4} \frac{(z^\beta + \delta)\; (1-z)^\gamma}{(\hat{z}^\beta + \delta)\; (1-\hat{z})^\gamma} \,, \tag{2.68}$$

where β, γ, δ, and $N_{3,4}$ ($\equiv g_{3,4}(\hat{z})$ with $\hat{z} = 0.5$) are free parameters.

There are a total of 11 parameters, 4 for the TMD PDF $(g_1, \alpha, \sigma, \lambda)$, 6 for the TMD FF $(g_3, \beta, \delta, \gamma, \lambda_F, g_4)$ and one for the nonperturbative part of the evolution (g_2), to be determined from data.

In principle, the parameters g_1, g_3, g_4, which determine the Gaussian width of the TMD distributions, may depend on the parton flavor q [80, 146, 147]: however, in the PV17 analysis they are assumed to be flavor independent since most of the data that are considered in the PV17 extraction are not sufficiently sensitive to flavor differences.

The TMD extraction of Ref. [84] is based on Monte Carlo replicas, and the best–fit values of the parameters of the functions in Eqs. (2.65)-(2.68) are reported in Tabs. 2.2 2.3, where the label 'Replica 105' refers to the replica that better approximates the average values computed considering all the 200 replicas. In particular, Replica 105 has been chosen as the replica whose parameter values are closer to their mean values. It does not have a specific statistical significance, but it is quoted for convenience, in case one wants to reproduce only one example.

	$b_{\max}$ [GeV^{-1}] (fixed)	$b_{\min}$ [GeV^{-1}] (fixed)	g_2 [GeV^2]
All replicas	$2e^{-\gamma_E}$	$2e^{-\gamma_E}/Q$	0.13 ± 0.01
Replica 105	$2e^{-\gamma_E}$	$2e^{-\gamma_E}/Q$	0.128

Table 2.2: Values of parameters common to TMD PDFs and TMD FFs.

TMD PDFs	g_1 [GeV^2]	α	σ		λ [GeV^{-2}]	
All replicas	0.28 ± 0.06	2.95 ± 0.05	0.17 ± 0.02		0.86 ± 0.78	
Replica 105	0.285	2.98	0.173		0.39	
TMD FFs	g_3 [GeV^2]	β	δ	γ	λ_F [GeV^{-2}]	g_4 [GeV^2]
All replicas	0.21 ± 0.02	1.65 ± 0.49	2.28 ± 0.46	0.14 ± 0.07	5.50 ± 1.23	0.13 ± 0.01
Replica 105	0.212	2.10	2.52	0.094	5.29	0.135

Table 2.3: 68% confidence intervals of best–fit values for the parameterizations in Eqs. (2.65)-(2.68) at $Q = 1$ GeV. We remark that these parameters have been obtained through a simultaneous fit of SIDIS and Drell–Yan data.

Chapter 3

NangaParbat: a framework of tools

We created a suite of tools, called `NangaParbat`, written in C++, that can be used for two main tasks: fitting unpolarized TMD PDFs on Drell-Yan data and generating grids for TMD PDFs, FFs and for the structure function $F_{\mathrm{UU,T}}(x, z, q_T, Q^2)$. The fit can be performed with different functional forms, but we will mainly refer here to the choice we adopted for our most recent TMD extraction [95], which will be denoted as PV19. The grids can be produced in principle from any TMD extraction compatible with NangaParbat prescriptions. We implemented grids based on the PV19 extraction and also on the previous extraction of Ref. [84], which will be denoted as PV17.

In this chapter we are going to describe the structure of `NangaParbat` and the implementation of its various functions, objects and integration tools.

The initial part of this chapter is dedicated to the description of the steps necessary to extract TMDs from data, while a second part will be devoted to explaining how the code to create grids works.

A stable version of the `NangaParbat` fitting framework is publicly available in the GitHub repository:

`https://github.com/MapCollaboration/NangaParbat`

In particular, the version of the first public release which we used to obtain results in Ref. [95] is *v1.4.0*. Updated instructions for the installation of `NangaParbat` are reported in the `README.md` available in the GitHub repository.

3.1 NangaParbat directory structure

The repository includes several directories, and some of them will be illustrated in the next sections. Apart from some folders related to more technical aspects of C++, like `bin/` and `inc/`, `NangaParbat` also contains:

- `cards/` - contains configuration files for the fit: their use will be explained in Sec. 3.4.
- `cli/` - stands for "command line interface". The programs in this directory have been conceived as `Python 3` wrappers, to simplify users access to `NangaParbat`. For example, running `fit.py` will help the user to set up and perform a TMD fit with the available datasets, as will be explained in Sec. 3.2.
- `doc/` - contains useful "how to" documents for the use of `NangaParbat` as well as some files, like `ConvTablesPrefactors.md` with information on the origin of prefactors included in the computation of the perturbative part of the extracted TMDs (discuss in Sec. 3.4).
- `FitResults/` - when downloading `NangaParbat`, in addition to the code, there are also has the raw results of the PV19 fit: in this folder the output of the code after the fitting procedure is stored. Note that, when a new fit is performed with `NangaParbat`, the output will have the same structure and format as the directory `FitResults/`, with one sub–folder for each replica generated in the extraction.
- `rawdata/` - contains data tables exactly as they can be downloaded from `HEPData` (`https://www.hepdata.net/`). The content of this folder will be discussed in Sec. 3.3, along with the directory `data/`.
- `run/` - contains the main codes needed for fitting and gird creation. Its content will be broadly explained across all sections in this chapter.
- `src/` - contains the source code is, the core of `NangaParbat` for calculations and function implementations. Also this directory will be cited throughout the course of this chapter.
- `tables/` - contains precomputed convolution tables for all orders of accuracy that we considered. They will be discussed in Sec. 3.4. They are created by `NangaParbat` and used in the fit.
- `tests/` - is a collection of stand–alone test programs.
- `tools/` - contains all the codes that perform convolutions and interpolations of TMD and structure functions grids.

For the sake of completeness, it should be noted that there is also the folder `resources/`, which, however, has nothing to do with the computational part of `NangaParbat`, since it contains tools to modify the code style and to visualize the progresses of the GitHub repository.

3.2 Fit with NangaParbat

To run a fit with `NangaParbat`, the user has two equivalent choices: the first option is to manually modify the appropriate configuration files[1] and then launch `run/RunFit`, which gets its input information from the terminal command line, as specified in the `help` section of the program.

The second option is more user–friendly: the command line interface `Python 3` program `cli/fit.py` allows a custom configuration of the fitting procedure through a series of pre–defined questions and launches the minimization with the selected setup.

3.2.1 Fit with command line interface: `cli/fit.py`

The program `fit.py` is designed to be an easy access to the code performing fits.

When starting, the user is asked to choose the name of the directory where the output files will be stored. Such directory must be new, not present in the main folder: [2] if a folder with that same name (and path) already exists, `fit.py` will be suspended, giving the user the possibility to change the name into one not already assigned to other directories. Making sure that the output of each fit will be put in a different folder is a useful feature to avoid overwriting results. When writing the results, everything related to the setup and the characteristics of the fit will be copied in the output folder, so that the output directory contains all the information to exactly reproduce the fit, leaving no ambiguities about specific settings.

Then, `fit.py` runs a filter program (`run/Filter.cc`) over the files present in `rawdata/` in order to produce data tables in the correct format to be read by `NangaParbat`. More details about this process are given in Sec. 3.3.1. In this second step the user can choose from a list which datasets to include in the fit. After this selection, `fit.py` writes the names of the designated datasets in `data/datasets.yaml`, which will serve as input when the actual fit will be launched at the end of the program. The data tables of the selected datasets are copied into the output folder.

The third step concerns the choice of the perturbative accuracy, which selects the precomputed convolution tables among those available in `tables/`. The characteristics of these tables, their meaning and how they are computed are discussed in Sec. 3.4.1 and also in Ch. 4. An overview of the characteristics of the selected tables is presented

[1] Specifically, those file are: the configuration file in `cards/` relative to the chosen parameterization (*i.e.* `fitPV19x.yaml` for PV19), `data/datasets.yaml`.

[2] If users want the output folder not to be at the main level of the NangaParbat directory, they can also give a relative path as answer to this first question. A folder corresponding only to the last name of the path will be created and not the whole path (meaning that if the given path is, for example, `not_here/but_here`, the directory `not_here/` has to exist prior to running `fit.py`).

to the user in the next step: the information shown is the content of the `config.yaml` file corresponding to the chosen perturbative accuracy. For example, if the user chooses N^{3}LL accuracy, such file is `tables/N3LL/config.yaml`.

At this point, users can confirm or deny their choice: in the latter case, `fit.py` stops and the user has to run it again from the beginning. Again, tables used for the fit are copied (or, in this case, symbolically linked, to save disk space[3]) in the output directory.

Next step of the fit setup is the choice of parameterization, which, as the previous features, can be done selecting the desired parameterization name from the automatically generated list. Such list is created by `fit.py` by running the program `run/AvailableParameterisations.cc`, which reads the map:

```
const std::map<std::string, Parameterisation*> AvPars
{
 {"DWS",    new NangaParbat::DWS{}},
 {"PV17",   new NangaParbat::PV17{}},
 {"PV19b",  new NangaParbat::PV19b{}},
 {"PV19x",  new NangaParbat::PV19x{}}
};
```

in `inc/NangaParbat/nonpertfuncions.h`.

The name of the parameterization of our latest fit [95] is *PV19x*. Parameterizations will be discussed in more details in Sec 3.4.2.

From here on, questions of the command line interface program are about the content of the main configuration file, which is stored in `cards/` and has the same name of the selected parameterization. Once the parameterization is chosen, a preview of its corresponding configuration file is shown: if users wish to start from a different default file, they have to indicate the relative path of the new configuration file.

For each question there is a default answer: if the user presses enter for each one of them, the fit runs with the same setup reported in the default configuration file, which is copied, unaltered, in the output folder.

Otherwise, users can modify any of the following:

- the text line to put as a description for the fit;
- whether or not to minimize: to skip minimization and have only predictions with the default parameters select *none*. It is worth mentioning here that is also possible to perform a scan in parameter space at the end of the minimization, to see if a (relative) minimum has really been reached: to use this option, select *scan* as choice of minimizer (it calls the `Root Minuit2` scanner function). More details on the characteristics of minimizers are discussed in Sec. 3.7;

[3] Our precomputed convolution tables have a variable size, which spaces from $\sim$ 300 KB to $\sim$ 25 MB.

- the random-number seed for the generation of the Monte Carlo replicas (to guarantee replicability of the results, using the same seed will give exactly the same results);
- the maximum value of q_T/Q allowed in the fit (see Ch. 4 for more details);
- the percentile cut [%] used for the report that can be generated with the program `cli/report.py` (for more details see Sec. 3.2.3);
- whether or not to use the t_0 prescription (see Sec. 3.6);
- whether or not to fluctuate the initial parameters, to explore a wider range in parameter space;
- the default starting parameters: the code presents the values of the default configuration file, but any of them can be changed;
- the default starting t_0 parameters (it works analogously to the previous point);
- the number of Monte Carlo replicas to run;
- whether to run the fit locally, on the machine used to run `cli/fit.py`, or on a Linux cluster with `SLURM system` (note that this second option has to be customized, since it is presently made for the cluster of the INFN Theory Group in Pavia). If the fits are launched on a cluster, they are run in parallel, otherwise they are performed one by one.

3.2.2 Set up and launch `run/RunFit`

To see what are the necessary inputs for `RunFit`, type `./RunFit --help`. Running `./Runfit` without the proper number of elements required in the input line will result in an automatic display of the help banner. To run a fit, users can also choose to manually configure the necessary files: in other words, they have to perform all the operations carried out by `cli/fit.py`.

The program `run/RunFit` requires five different inputs form the command line on the terminal:

```
1 ./RunFit <output dir> <fit configuration file> <path to data
      folder> <path to tables folder> <replica ID>
```

Unlike for `cli/fit.py`, the output folder `<output dir>` has to exist before launching `./RunFit`.

The main information that guides the choice of `<fit configuration file>` is the parameterization. Some configuration file examples are located in `cards/` and are

named after the parameterization they contain. From those files `run/RunFit` reads some of the key features of the fit: the minimizer; the seed used with the random–number generator for the generation of the Monte Carlo replicas; the cut on q_T/Q;[4] the percentile cut on the distribution of χ^2s, error functions and parameters used to identify outliers when running `cli/report.py`; whether to use the t_0 prescription and with what parameters, the name of the parameterization, the values of the relative parameters and whether or not to fluctuate them.

Besides the parameterization of the nonperturbative part that we selected for our TMD extraction in Ref. [95], *PV19x*, in `NangaParbat` a few other possible choices are implemented, with different functional forms. It is also possible to create new parameterizations, to expand and improve the fit, and to extract also other TMDs (for example, the Sivers TMD). To do so, one can follow the steps described at the end of Sec. 3.4.2.

The requested path to the data folder is the relative path of the data directory with respect to where the user is running `RunFit.cc` from. For example, to reproduce the same results of [95], or simply to use the same datasets, `<path to data folder>` has to be the path to `data/`. Data selection is done based on the content of `data/datasets.yaml`: all the datasets listed in such file are going to be included in the fit.

When running `run/RunFit`, without using `cli/fit.py`, the program `run/Filter` is not executed and the fit uses directly the tables stored in `data/`, without considering `rawdata/`. In the end, this should not matter, as the result of running the filter on rawdata to produce data tables in the right format for NangaParbat is always the same. However, to be on the safe side and to avoid the use of data files potentially corrupted by accident, it is advisable run `run/Filter.cc` every time before launching a fit.

As for `<path to tables folder>`, this is the relative path to the directory where precomputed convolution tables are stored. For example, to perform a fit with a perturbative accuracy of N^{3}LL, such path has to be the path to `tables/N3LL`.

Every sub–directory of `tables/` contains `config.yaml`, which specifies important features of the convolution tables in that sub–directory. Among the features in `config.yaml` there are: the set of collinear PDFs and (if needed) the set of collinear FFs, the name of the b_* prescription (see Ch. 2), the perturbative order of the computation, the initial and final scale–variation factors around $\mu_b = 2e^{-\gamma_E}/b$ and Q, the reference value of $\alpha_{\rm em}$ (the electromagnetic coupling constant), the maximum value allowed for the ratio q_T/Q for the computation of the tables (which must be greater or equal to the value of such q_T/Q cut in the fit).

[4] This has to be smaller than the production cut used to produce the precomputed convolution tables, which is $q_T/Q < 0.3$, as can be seen in every `config.yaml` file in each sub–directory of `tables/`.

Entries of `config.yaml` should not be changed: they describe with what configuration the perturbative part was calculated. For example, if the tables were precomputed using the `MMHT2014nnlo68cl` collinear PDFs set, the fit should be consistently done with the same set. We chose to precompute tables with perturbative terms in order to speed up the computation. Such tables are calculated for every dataset, and the advantage of producing them is that the computation is done once and for all. However, one disadvantage of this method is that, whenever the user wants to change one of the features in the `config.yaml` file described above, all the tables have to be computed again. It is also possible to customize the convolution tables contained in `tables/`, by changing the desired features in the `config.yaml` relative to the requested perturbative accuracy and then running the program `cli/tables.py`, which uses `run/CreateTables.cc` to produce new convolution tables. More details about the computation of the convolution tables are given in Sec. 3.4.1.

Finally, the last input required by `./RunFit` is a number to use as the identification of the replica associated with the fit that is performed. As `<replica ID>` the user has to provide an integer number. Contrary to what happens in `cli/fit.py`, where the number provided by the user is the amount of fits that the user wants to run, here the `<replica ID>` serves (also) the purpose of labeling for the result of the fit:[5] when using `run/RunFit`, is not possible to launch multiple fits, the user can perform the fit for only one replica at a time.

3.2.3 Create a report from NangaParbat output using `cli/report.py`

The command line interface program `cli/report.py` produces a report of the fit results containing a general statistical analysis of the extractions obtained after running a fit and some plots that are particularly meaningful for our studies.

When running `cli/report.py`, the first element needed by the program is the path of a NangaParbat fit output folder. Such folder is assumed to have the structure illustrated in Fig. 3.1. Highlighted files and folders are the ones which are actually used by `report.py`: in particular, note that there must be a directory named "replica_0" and that there must be more than only one replica present in the output folder.

Then, `cli/report.py` selects the replicas that can be used for statistical analysis. A first step in the replica selection is done by looking at the status indicator in each of the replica reports (`/Report.yaml`): only replicas that converged[6] (`Status: 1`) are

[5] The `<replica ID>` is also used as a proper integer number in functions and classes present in `NangaParbat`, in particular in `src/data/datahandler.cc`. More details about it are given in Sec. 3.3.2.

[6] The convergence of a replica is determined by the algorithms of the selected minimizer

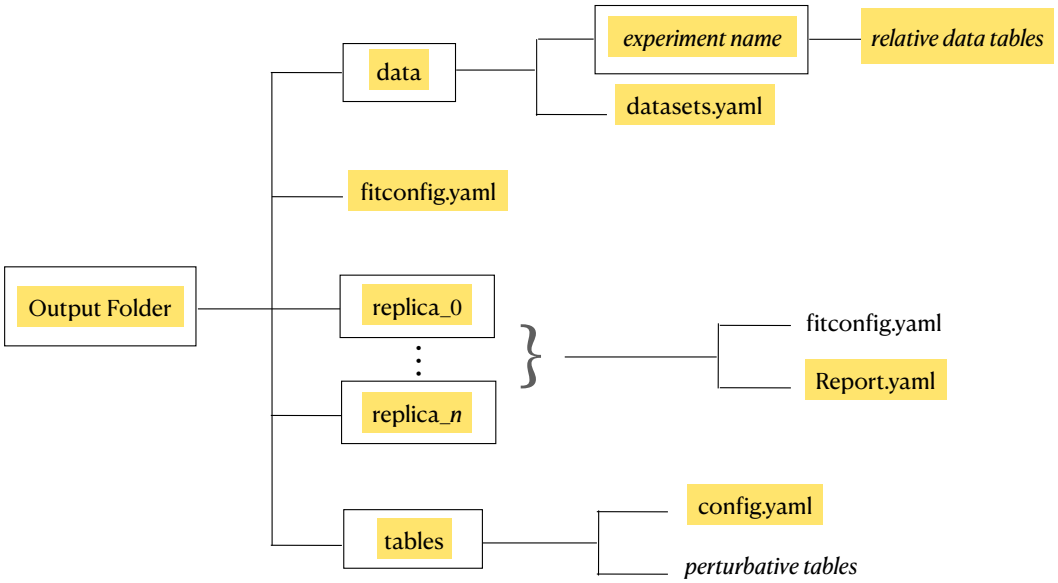


Figure 3.1: Structure of a `NangaParbat` output. Names in boxes refer to directories, whereas names without boxes are simple files; italic text generically describes the content at that level of the structure, while labels not in italic are literal expressions needed for such directories and files. Finally, highlighted files and directories indicate the elements for which `cli/report.py` explicitly searches: if those items are not in the right place or do not exist, `cli/report.py` stops.

considered in the following steps of the study. An update on the discarded replicas is printed on terminal: for each of them the identification number and their global χ^2 are indicated.

As a further step in the replica selection, depending on which percentage has been indicated as `Percentile cut` in `fitconfig.yaml` certain replicas are identified as outliers according to the χ^2, error function, and parameters distributions and they are removed from the list of usable replicas. For example, if `Percentile cut: 5`, then all the replicas whose global χ^2 or error function or (even only) one parameter value belongs to the lowest 5% or the highest 5% of their respective distributions are discarded. The percentile cut is chosen arbitrarily, following a rule of thumb that suggests to eliminate replicas with very high or very low χ^2, which is likely due to numerical fluctuations in the minimization. Moreover, when choosing the number to use as a threshold, one has to keep in mind that a very high percentile might discard useful replicas in the tail of the χ^2 distribution, while if the value of the percentile cut is too small it might not be effective in excluding outliers.

In the next step the mean replica is calculated, since part of the resulting analysis will be based on comparisons of the behavior of the replicas within respect to their average. To do this, `cli/report.py` launches `run/ComputeMeanReplica`, which uses the code in `src/parameterisation/meanreplica.cc`: the mean replica is computed point by point, averaging the values of all the replicas in the kinematic points corresponding to the data of each experiment.

We stress that the mean replica is created numerically, it does not correspond to a specific set of parameters, but it is a valid figure of merit to evaluate fit results, as discussed in Ch. 4, where some of the features of the statistical analysis performed by `cli/report.py` are described.

The code `run/ComputeMeanReplica`, once called, also computes the χ^2 of the mean replica (without minimization) and produces a directory, `mean_replica/`, with the relative `Report.yaml` in it.

Then, the subsequent part of `cli/report.py` is dedicated to the production of plots, some of which are commented in Ch. 4. The following figures are created: a plot for the global χ^2 distribution, one for the error function distribution and one for the distribution of each of the fitted parameters; a colored table to show correlations between parameters; three plots of TMDs in momentum space in three different kinematic regions; one plot for each experiment, displaying predictions compared with data points and also, on another panel, the χ^2 distribution for that particular dataset.

As part of the statistical analysis, the report also includes tables of the single experiments χ^2 distributions of the central replica (replica_0, the one obtained without

(those available in `NangaParbat` are `Minuit2` and `ceres-solver`).

data fluctuations), the mean replica and the average over the replicas. The distributions of the single parameters, with their mean value and standard deviations, are also reported.

At the end of the analysis, the final report is available in the directory `Report/`, created by `cli/report.py` inside the fit output folder, in various formats (`markdown`, `pdf`, `HTML` and LaTeX)[7].

3.3 Data preprocessing

Before any fit, data tables have to be generated. Starting from the official data of the various experiments, available to the community through `HEPdata`, `NangaParbat` is able to produce data tables adapted to the needs of the fitting procedure through a program that acts like a filter.

Ideally, the procedure that converts original data tables (rawdata) to data tables that can be used in `NangaParbat` should be done every time a fit is performed: in this way there is the certainty that no human error can enter the data tables that serve as a base for the fit.

In the next section, the data filter is described and the procedure to add new data and make them available for a `NangaParbat` fit is explained.

3.3.1 Filter

The data filter consists of a collection of codes, run by `run/Filter.cc`, which from the content of `rawdata/` produces from scratch another directory, `data/`, with the `NangaParbat` data tables, stored in `src/preprocessing/`.

Each code in `src/preprocessing/` is named after the experiment it refers to, and each experiment has its own tailored function that transforms the original data table (or data tables, as there can be more than one set of data available for a single experiment) into the corresponding `NangaParbat` data table. The output can consist in more than one table, depending on the characteristics of the data, *i.e.*, the presence of a binning in rapidity calls for separate tables, one for each bin.

The structure of the data filter can be described in the following way: each `src/preprocessing/preprocessEXPERIMENT.cc` contains a function, named `PreprocessEXPERIMENT`, that is called by the program `run/Filter.cc`. Such program takes as inputs the path to rawdata folder (`rawdata/`) and the path where the folder

[7]The conversion from `markdown` to the other formats is done through a tool called `Pandoc`, whose files are copied into the final report folder by `cli/report.py` itself.

of the processed data will be placed (`data/`).[8]

As already mentioned before, the output of `run/Filter` are data tables in `yaml` format, with the correct headers and variables to be read by `NangaParbat` itself, plus a file, `data/datasets.yaml`, which contains the list of all the available datasets that have been processed. All the datasets included in `data/datasets.yaml` are considered when performing a fit (either with `run/RunFit` or with `cli/fit.py`).

The datasets available in `NangaParbat` *v1.4.0* are those discussed in Ch. 4.

In order to add new datasets, the following steps are necessary:

- download the new data folder from `HEPData` and put it in `rawdata/`;
- create a file in `src/preprocessing/`, *i.e.*, `preprocessingNEWEXP.cc`, and add its name in `src/preprocessing/CMakeLists.txt`;
- encode in `preprocessingNEWEXP.cc` the function that reads rawdata and creates the appropriate `NangaParbat` datasets;
- create the function `PreprocessNEWEXP` in `inc/NangaParbat/preprocessing.h`, following the structure of other similar functions already present there, such as:

```
std::string PreprocessATLAS8TeV(std::string const&
   RawDataPath, std::string const& ProcessedDataPath, bool
    const& PDFError = true);
```

- add in `run/Filter.cc` the call to the newly created function;
- re–compile and install the `NangaParbat` library;
- run `run/Filter` to produce new data tables in the correct format.

3.3.2 Data Handler

Data tables are read by objects of the `NangaParbat::DataHandler` class, which is defined in `inc/NangaParbat/datahandler.h` and encoded in `src/data/datahandler.cc`. The DataHandler class provides a common interface to all datasets: it provides methods to get kinematics, central values, uncertainties and labels used for plotting and to compute and handle the covariance matrix, its Cholesky decomposition, the fluctuation of the initial parameters of a fit and the use of t_0 parameters (see Sec. 3.6 for more information regarding these last two functionalities).

[8] The directory of the processed data is created each time, so in principle is not necessary to call it `data/`. When `cli/fit.py` is used, the filter program is automatically run with `os.system(RunFolder + "/Filter " + RunFolder + "/../rawdata/ " + outfolder + "/data")`.

From the point of view of the structure of `NangaParbat`, one of the two elements that are needed for the χ^2 computation has to belong to the DataHandler class. This can be seen, for example, in the last part of `run/RunFit.cc`:

```
// Open datasets.yaml file that contains the list of datasets
// to be fitted and push the corresponding pairs of
// "DataHandler" and "ConvolutionTable" objects into the a
   vector.
const YAML::Node datasets = YAML::LoadFile(std::string(argv
   [3]) + "/datasets.yaml");
for (auto const& exp : datasets)
  for (auto const& ds : exp.second)
    {
      std::cout << "Reading table for " << ds["name"].as<std::
         string>() << "..." << std::endl;

      // Convolution table
      const std::string table = std::string(argv[4]) + "/" +
         ds["name"].as<std::string>() + ".yaml";
      const NangaParbat::ConvolutionTable ct{YAML::LoadFile(
         table), fitconfig["qToQmax"].as<double>()};
      //ct.NumericalAccuracy(NPFunc->Function());

      // Datafile
      const std::string datafile = std::string(argv[3]) + "/"
         + exp.first.as<std::string>() + "/" + ds["file"].as<
         std::string>();
      const NangaParbat::DataHandler dh{ds["name"].as<std::
         string>(), YAML::LoadFile(datafile), rng, ReplicaID,
                                        (fitconfig["
                                           t0prescription"].as
                                           <bool>() ? ct.
                                           GetPredictions(
                                           NPFunc->Function())
                                            : std::vector<
                                           double>{})};

      // Add chi2 block
      chi2.AddBlock(std::make_pair(dh, ct));
    }
```

DataHandler acts as a bridge between data tables and χ^2 computation: `src/data/datahandler.cc` encodes the way to properly read the uncertainties taken into account in the computation of the covariance matrix, which is also done in this part of the code. More details about the meaning of the different types of uncertainties and their role in the calculations of the covariance matrix can be found in Sec. 3.6.

The Datahandler constructor, encoded in `inc/NangaParbat/datahandler.h`, is:

```
DataHandler(std::string const& name, YAML::Node const&
   datafile, gsl_rng* rng = NULL, int const& fluctuation = 0,
   std::vector<double> const& t0 = {});
```

It takes five arguments: the the name associated to the data set, the `YAML::Node` with the interpolation table, a GSL random number object, the ID of the fluctuation (*i.e.*, Monte-Carlo replica ID, 0 means no fluctuations) and a vector of predictions to be used for the t_0 prescription. In particular, the vector with the t_0 coefficients is needed for the computation of the covariance matrix, as explained in Sec. 3.6.2.

DataHandler is also responsible for the fluctuation of initial data. If the user chooses to run N_{rep} replicas of the fit, with $N_{rep} > 1$, or if the user would like to have a TMD extraction corresponding to a specific replica ID, then DataHandler fluctuates each dataset and creates an ensemble of pseudodata to fit.[9] Such pseudodata are univocally linked to the replica identification number, to guarantee that a fit for a certain replica ID will always give the same result when there are the same initial conditions, even if random fluctuations of data are involved in the process of generating the pseudodata to fit. This is achieved using the replica ID and the random–number generator `gsl_rng * rng` initialized with the seed number provided in the initial fit settings.

In the call to the DataHandler constructor, `int const& fluctuation` is set equal to the replica ID. Then, for each dataset, the full dataset is fluctuated `fluctuation` times, and only the last fluctuation is kept. This is not efficient but, as mentioned above, it serves the purpose of identifying a given random replica by its ID and the random seed given in the initial fit settings.

The statistical sample of N_{rep} replicas is obtained by generating artificial data points following a Gaussian distribution centered on each data point with a variance given by the experimental uncertainty.

To determine the standard deviation of the Gaussian used for the random shift of each point, the implemented formulas take into account all uncertainties, according to their nature. The technique used is inspired by the one reported in Ref. [169]. For each dataset, the fluctuated i-th data point for the k-th replica, $F_i^{(k)}$, is given by:

$$F_i^{(k)} = F_{i,\text{mult}}^{(k)} \cdot m_i \left(1 + F_{i,\text{add}}^{(k)} + F_{i,\text{unc}}^{(k)}\right) \tag{3.1}$$

where m_i is the measured experimental value (the original, non fluctuated i-th point in the dataset), $F_{\text{mult}}^{(k)}$ is the multiplicative correlated uncertainty fluctuation, $F_{\text{add}}^{(k)}$ the additive correlated uncertainty fluctuation and $F_{\text{unc}}^{(k)}$ the uncorrelated uncertainty fluc-

[9] If, with `cli/fit.py`, the fit is launched for only one replica, then the the ID of such replica is 0 and the only replica produced is replica_0, also called *central replica* in the analysis done by `cli/report.py`

tuation. The correlated contributions to the fluctuations are:

$$F^{(k)}_{i,\text{mult}} = \prod_{l}^{N_m} \left(1 + \delta^{(l)}_{i,\text{mult}} r^{(k)}_{l,\text{mult}}\right) \qquad F^{(k)}_{i,\text{add}} = \sum_{l}^{N_a} \delta^{(l)}_{i,\text{add}} r^{(k)}_{l,\text{add}} \tag{3.2}$$

where N_m (and N_a) are the numbers of multiplicative (or additive) uncertainties for each experimental point,[10] $\delta^{(l)}_{i,\text{mult}}$ and $\delta^{(l)}_{i,\text{add}}$ are the relative multiplicative and additive uncertainties associated with the i-th data point and $r^{(k)}_{l,\text{mult}}$ and $r^{(k)}_{l,\text{add}}$ are univariate Gaussian random numbers, *i.e.*, random numbers extracted from a Gaussian distribution with unitary standard deviation and mean equal to zero, $N(\mu = 0, \sigma = 1)$.

The uncorrelated contribution $F^{(k)}_{i,\text{unc}}$ is:

$$F^{(k)}_{i,\text{unc}} = \frac{r^{(k)} s_i}{m_i} \tag{3.3}$$

where s_i is the sum in quadrature of all the uncorrelated uncertainties associated with the data point m_i and, as before, $r^{(k)}$ is a number, randomly extracted from a standard Gaussian distribution ($N(\mu = 0, \sigma = 1)$), associated with the replica identification number k. Since s_i are the absolute values of the uncorrelated uncertainties and not the relative values as it happens for the multiplicative and additive uncertainties, denoted with $\delta^{(l)}_{i,\text{add}}$, in Eq. (3.3) s_i is divided by the experimental measurement m_i (see also Sec. 3.6.2).

For each replica k, if two experimental points have the same correlated systematic uncertainties, the fluctuations due to the correlated systematic uncertainties will be the same for both points. A similar statement can be made for normalization uncertainties, which are are properly taken into account in $F^{(k)}_{i,\text{mult}}$.

One of the tasks of DataHandler is also preparing inputs for integrations: it reads the bin boundaries for every variable involved in the process and stores them for later use in the integrals to build the observable. DataHandler is organized to take into account integrations in q_T, and other two or three variables, depending on the process (DY or SIDIS): the invariant–mass (for DY) or virtuality (for SIDIS) Q, the rapidity y (for DY) or the Bjorken–x x (for SIDIS) and z (only for SIDIS).

Each of these variables, in addition to upper and lower values for the bins, also has a boolean flag associated with it. For example, if in the header of the data table there is `integrate: true` such variable is integrated over, otherwise, if `integrate : false`, then `NangaParbat` considers the mean value of the variable, calculated as the arithmetic average between the upper and lower bound of the bins. For q_T, if the

[10] A data point can have more than one uncertainty per type: for example, it can have two different additive errors and one normalization error, and in this case, $N_m = 1$ and $N_a = 2$.

keys `low` and `high` are present, the data point is integrated over the q_T bin.

3.4 Building the observable

In Ch. 2 we described the theoretical framework to obtain the results in Ref. [95]. The main purpose of this section is to illustrate how such theoretical framework is implemented in the structure of `NangaParbat`.

For greater clarity, we restate here the formula written in Eq. (2.10), which expresses the fully differential cross section for lepton–pair production in the region where it is possible to apply TMD factorization, *i.e.* $q_T \ll Q$:

$$\begin{aligned}\frac{d\sigma}{dQdydq_T} &= \frac{8\pi\alpha_{\rm em}^2 q_T \mathcal{P}}{9Q^3} H(Q,\mu) \\ &\times \sum_q c_q(Q) \int_0^\infty db_T\, b_T\, J_0\big(b_T q_T\big)\, x_1 \tilde{f}_1^q\big(x_1, b_T; \mu, \zeta_1\big)\, x_2 \tilde{f}_1^{\bar{q}}\big(x_2, b_T; \mu, \zeta_2\big)\ ,\end{aligned} \tag{3.4}$$

where Q, y, and q_T are the invariant mass, rapidity, the transverse momentum of the lepton pair, respectively, while α is the electromagnetic coupling, H is the appropriate QCD hard factor, c_q are the effective electroweak charges, and $\mathcal{P}$ is the phase–space reduction factor, which is discussed in Appendix 3.A.

Considering Eq. (2.18) and Eq. (2.39), where the b_* prescription is introduced (see also Sec. 2.1.5), the expression for the TMD PDF $\tilde{f}_1(x, b_T; \mu)$ in b_T–space can be schematically written as

$$\tilde{f}_1^i(x, b_T; \mu) = \sum_{j=g,q,\bar{q}} \big(C_{ij} \otimes f^j\big)\,(x, b_*; \mu_b) e^{R(b_*;\mu_b,\mu)} f_{\rm NP}(x, b_T). \tag{3.5}$$

where C_{ij} $(i = q, \bar{q})$ are the matching functions to the collinear PDFs f^j, $e^{R(b_*;\mu_b,\mu)}$ is the Sudakov form factor (see Sec. 2.1.3) and $f_{\rm NP}(x, b_T)$ is a phenomenological function that is introduced to account for nonperturbative effects that become relevant at large b_T in TMD evolution and matching. In Eq. (3.4) and (3.5), the hard factor, the matching functions and the Sudakov factor can be perturbatively computed, while $f_{\rm NP}$ represents the nonperturbative TMD components of the unpolarized TMD $\tilde{f}_1^i$ and has to be extracted from data.

As it has been shown in Ch. 2, Eq. (2.39, when extracting TMDs two contributions can be identified, perturbative and nonperturbative. This section is divided in two parts: Sec. 3.4.1 is dedicated to terms that can be precomputed in a TMD extraction, while Sec. 3.4.2 describes the implementation in `NangaParbat` of the nonperturbative function $f_{\rm NP}$.

As the calculation of the terms that can be precomputed heavily relies on integration techniques, some of the formulas that are actually implemented in our code are discussed in a separate section, Sec. 3.5, dedicated to integrations in `NangaParbat`.

3.4.1 Precomputed terms: convolution tables

In order to extract the unpolarized TMD $\tilde{f}_1(x, b_T; \mu, \zeta)$ from data, a fitting procedure is necessary: this requires multiple calls to the computation of the observable, which in this case is the cross section given by Eq. (2.10). Since during the fit the cross section has to be evaluated many times (up to thousands), using the expression in Eq. (2.10) every time to compute the observable turns out to be an highly expensive task in terms of computational time and resources.

It is thus optimal to use an approach that separates the computation of known parts from the nonperturbative terms that need to be extracted. Therefore, in order to speed up the fit of the nonperturbative function $f_{\rm NP}$, Eq. (2.39), we make use of interpolation techniques inspired by those employed for collinear–factorization predictions [170–174]. For instance, programs like `FastNLO` [175], `APPLgrid`[172] and `APFELgrid` [176] are used for collinear PDF fits, [11] and demonstrate how to separate perturbative and nonperturbative contributions. Within this approach, all the terms that can be precomputed are calculated once and for all and then stored in suitable tables.

The explicit formulas given in Sec. 3.5 show how the transverse nonperturbative terms can be decoupled from other contributions during the computation of the cross section in Eq. (3.4).

Schematically, in our TMD study, we reduce the computation of the cross section in Eq. (3.18) for a given kinematic bin to the weighted sum

$$\frac{d\sigma}{dq_T} \simeq \sum_{n,\alpha,\tau} W_{n\alpha\tau} f_{\rm NP}(x_1^{(\alpha,\tau)}, b_T^{(n)}, \zeta^{(\tau)}) f_{\rm NP}(x_2^{(\alpha,\tau)}, b_T^{(n)}, \zeta^{(\tau)}) \,, \tag{3.6}$$

where the discrete variables $x_{1,2}^{(\alpha,\tau)}$, $b_T^{(n)}$, and $\zeta^{(\tau)}$ run over appropriately defined grids. Specifically, the index n refers to the points of the Ogata quadrature method [180], used to compute integrals needed to obtain the weights W (see also Sec. 3.5.2), while α and τ run respectively over x and Q grids.

The computationally expensive part of the calculation is isolated into the weights $W_{n\alpha\tau}$ that are precomputed and stored in `yaml` tables. Such tables, called *convolution tables* in this work, are stored in `tables/`: there is a folder for each logarithmic order (LL, NLL, NLL′, NNLL, NNLL′, N^{3}LL) and each of these directories contains a table

[11] `FastNLO` is used by various global PDF fitting groups, like MSTW [177], CTEQ [178], while NNPDF [179] relies on `APFELgrid` and `APPLgrid`.

for each dataset that is possible to include in a `NangaParbat` fit.

This procedure makes the computation of predictions very fast and thus suitable for a fit that requires a large number of iterations.

To compute the `NangaParbat` convolution tables the `Python` program, `cli/tables.py` configures and runs `run/CreateTables` (that can also be executed independently from the command line interface script, in a similar way to `run/RunFit`.

Contrary to what happens for the generation of the `NangaParbat` data tables (see Sec. 3.3), the production of convolution tables has to be done only once, before starting the fit.[12]

Among the initial inputs of `run/CreateTables` there is a configuration file, `config.yaml`, which specifies the characteristics necessary to compute the weights $W_{n\alpha\tau}$. Such configuration file serves as input to allocate and initialize a *FastInterface* object, as the *FastInterface* constructor needs an input card, `config.yaml`, from which it reads the parameters to compute the convolution terms at the desired level of accuracy. For example, this is what `tables/N3LL/config.yaml` looks like:

```
# Collinear PDF set and member to be used for the generation
# of the tables. They are assumed to be in the LHAPDF format.
pdfset:
  name: "MMHT2014nnlo68cl"
  member: 0

# Collinear FF set and member to be used for the generation
# of the tables. They are assumed to be in the LHAPDF format.
ffset:
  name: "DSS14_NLO_PiSum"
  member: 0

# Name of the b* prescription. This name has to correspond to
# a function in inc/NangaParbat/bstar.h
bstar: "bstarmin"

# Perturbative order of the computation. 0: LL, 1: NLL,
# 2: NNLL, 3: NNNLL, -1: NLL', -2: NNLL'
PerturbativeOrder: 3

# Initial and final scale-variation factors around
mub = 2e^{-gamma_E} / b and Q, respectively.
TMDscales:
  Ci: 1
  Cf: 1
```

[12] It is suggested to generate data tables every time, but it is not mandatory to do so. Viceversa, it should be avoided to produce convolution tables more than once, because they are extremely time–consuming.

```yaml

# Reference value of alpha_em.
alphaem:
  aref: 0.00776578395589
  Qref: 91.1876
  run: true

# Parameters of the x-space subgrids on which collinear PDFs
# are tabulated.
xgridpdf:
  - [60, 1e-4, 3]
  - [60, 1e-1, 3]
  - [50, 6e-1, 3]
  - [50, 8e-1, 3]

# Parameters of the x-space subgrids on which collinear FFs
# are tabulated.
xgridff:
  - [60, 1e-2, 3]
  - [50, 6e-1, 3]
  - [50, 8e-1, 3]

# Maximum number of Ogata-quadrature points (no more than
   1000).
nOgata: 200

# Number of points, interpolation degree, and integration
# accuracy of the grid in Q.
Qgrid:
  n: 10
  InterDegree: 3
  eps: 1e-3

# Number of points, interpolation degree, and integration
# accuracy of the grid in xi = exp(y) or xF
# (depending on the observable).
xigrid:
  n: 10
  InterDegree: 3
  eps: 1e-3

# Maximum value allowed for the ratio qT / Q.
# This has to be intended as a "generation-level" cut meaning
# that only data points with qT / Q below this value are
# generated. Those above are simply set to zero.
# However, a more restrictive cut can still be applied
# at the fit level.
```

```
# This avoids computing predictions for points whose
# kinematics is such that TMD factorisation is not valid.
qToverQmax: 0.3
```

Note that, the configuration file also specifies the collinear PDF sets, which are treated as inputs in our TMD fit and needed to compute the convolution with the matching coefficients, $\sum_j \left(C_{ij} \otimes f^j\right)(x, b_*; \mu_b)$, in Eq. (3.5).

The version *v1.4.0* of `NangaParbat` can analyze only Drell–Yan data and therefore only TMD PDFs can be extracted. For this reason, only the collinear PDF set `MMHT2014nnlo68cl` plays a role in the production of convolution tables[13]. So, precomputed tables also depend on the chosen sets of collinear PDFs and FFs, and, in case a study or a fit with different PDFs (or FFs) has to be performed, the convolution tables have to be computed again.

In order to control the behavior of the TMDs at large b_T, a b_* prescription is introduced, as discussed in Ch. 2 Sec. 2.1.5. In particular, our convolution tables are computed with the so–called *bstarmin* prescription, which is given by Eq. (2.36)and is implemented in `src/utilities/bstar.cc`.

In `NangaParbat` it is also possible to use different prescriptions to deal with the high– and low–b_T region: once the functional form has been implemented in `src/utilities/bstar.cc` (and in `inc/NangaParbat/bstar.h`), the selection of the b_* prescription to use in the fit is done in `config.yaml`. At the moment, beside the *bstarmin* prescription, also a function à la Collins–Soper–Sterman (CSS) [98] is implemented[14].

The computation of the convolution tables relies on `APFEL++` [181, 182], a PDF evolution library in C++, and is done in `src/fastinterface/fastinterface.cc`, while `src/fastinterface/convolutiontable.cc` contains functions and methods to read such tables and make predictions.

In particular, `run/CreateTables.cc` calls the function `FastInterface::ComputeTables`, whose header is reported here from `inc/NangaParbat/fastinterface.h`:

```
/**
 * @brief Function that computes the interpolation tables
 * given as an input vector of "DataHandler" objects.
 * @param DHVect: vector of "DataHandler" objects.
 * @return a vector of "YAML::Emitter" objects containing
 * as many tables as elements of "DHVect".
 */
```

[13] However, even if it is not used, an entry for `ffset:` is required bt `NangaParbat`.

[14] The CSS prescription implemented in `NangaParbat` as *bstarCSS* is $b_* = b/\sqrt{1 + (b/b_{\max})^2}$.

```
8 std::vector<YAML::Emitter> ComputeTables(std::vector<
    DataHandler> const& DHVect) const;
```

The argument of this function is a DataHandler object (see Sec. 3.3.2) that contains the datasets for which convolution tables have to be produced.

Since the main goal of `FastInterface::ComputeTables` is to store in tables all the elements that concur to the calculation of the cross section in Eq. (3.4) and that can be precomputed, also the hard factor $H(Q, \mu)$ is taken into account. According to the information written in data files, the hard factor and the proper prefactors with the correct electroweak charges for the process are considered in the computation of the weights: convolution tables are therefore process–dependent. In particular, the hard factor is called from `APFEL++` by the constructor of the *FastInterface* class, and is used in `src/fastinterface/fastinterface.cc` in the functions `FastInterface::LuminosityDY` or `FastInterface::LuminositySIDIS`, which are called by `FastInterface::ComputeTables` accordingly to the process to study.

Moreover, in `FastInterface::ComputeTables`, depending on the indications written in the datafile regarding kinematic variables (see Sec 3.3), there is the possibility to choose whether to integrate over the bins or to consider only the mean values of x, Q and q_T.

If the selected choice is to integrate over the bins, such task is performed exploiting the the techniques explained in Sec. 3.5. For our fit in Ref. [95], convolution tables have been computed integrating over all the variables, x, Q and q_T.

Although to integrate over all the possible kinematic bins is the recommended choice, in some cases the possibility to perform calculations using the mean values (average over the bin, calculated with $(Q_{\rm max} - Q_{\rm min})/2$, for example) can come in handy. For example, in order to estimate the uncertainties given by the collinear PDFs using the Hessian method, one convolution table for each member of the PDF set has to be computed (for all the datasets). Such task is very time–consuming and therefore, for the case of Z–boson production, we produced convolution tables using the narrow–width approximation, which implies not integrating over Q but considering only $Q = M_{\rm Z}$. More details about the narrow–width approximation can be found in Appendix 3.C.

The function `FastInterface::ComputeTables` takes also into account the phase–space reduction factor $\mathcal{P}$ (see Appendix 3.A) that appears among the prefactors in Eq. (3.4). Since such prefactor is q_T dependent, it can not be completely factored out of the integration over q_T bins: however, making use of considerations and approximations explained in Sec. 3.5.1, a clever workaround to efficiently compute integrals with the presence of kinematic cuts on the leptons final states can be found.

Through the use of Q and x grids, specified with the entries `Qgrid` and `xigrid` in `config.yaml` and with the Ogata parameters also included in the configuration

file under nOgata, the weights W are calculated and written in specific `yaml` tables. The formulas that are actually implemented in our code and that exploit the Ogata quadrature method for integrations are discussed in Sec. 3.5.2.

The *ConvolutionTable* class (declared in `inc/NangaParbat/convolutiontable.h`) implements the methods for the numerical convolution of the interpolation tables with user–given nonperturbative functions.

Making use of convolution tables, predictions are obtained in `src/fastinterface/convolutiontable.cc` through the function `GetPredictions`

```
/**
 * @brief This function returns a vector of predictions given
 * a single user-given non-perturbative function.
 * @param fNP: the non-perturbative input functions
    parameterised by an index
 * @return a vector of predictions.
 */
virtual std::vector<double> GetPredictions(std::function<
   double(double const&, double const&, double const&)> const&
    fNP) const;
```

which calls the function `Convolute`:

```
std::map<double, double> ConvolutionTable::Convolute(std::
   function<double(double const&, double const&, double const
   &)> const& fNP1, std::function<double(double const&, double
    const&, double const&)> const& fNP2) const
```

which implements Eq. (3.40) and takes as argument the nonperturbative functions to fit, $f_{\rm NP}$.

At the end of `run/CreateTables.cc` there is also the possibility to check the quality of the convolution tables that have been produced in the first part of the program. Such accuracy control is triggered by an argument in the terminal line used to run the program:

```
./CreateTables <configuration file> <path to data folder> <
   output folder> <test tables? [y/n]> [optional selected
   datasets]
```

If the boolean fourth entry of the command line is `true`, predictions obtained with an interpolation of the convolution tables are compared with the result of a direct calculation. Direct computation is implemented in `src/fastinterface/fastinterface.cc`, in

```
/**
 * @brief Function that computes the predictions corresponding
     to
 * a set of "DataHandler" objects as direct integration.
```

```
4  * @param DHVect: vector of "DataHandler" objects.
5  * @param fNP1: the first non-perturbative input function
6  * @param fNP2: the second non-perturbative input function
7  * @param epsQ: the integration accuracy in Q
8  * @param epsxi: the integration accuracy in xi
9  * @param sameOgata: whether to use the same number of Ogata
       points or reach the required accuracy
10 * @return a vector of vectors, each containing the
       predictions
11 * associated to the corresponding "DataHandler" object
12 */
13 std::vector<std::vector<double>> DirectComputation(
14 std::vector<DataHandler>
                                                    const& DHVect,
15 std::function<double(double const&, double const&, double
       const&)> const& fNP1,
16 std::function<double(double const&, double const&, double
       const&)> const& fNP2,
17 double          const& epsQ = 1e-7,
18 double          const& epsxi = 1e-7,
19 bool            const& sameOgata = true) const;
```

3.4.2 Nonperturbative part: parameterization

The function $f_{\rm NP}$, in Eq. (3.5), has to be extracted from data, and to do this a functional form depending on a set of parameters has to be chosen. Many different parameterizations can be proposed, and our choice for $f_{\rm NP}$ is expressed in Eq. (2.42) and (2.43), that have been discussed in Ch. 2, Sec. 2.1.5.

In `NangaParbat`, we called such parameterization *PV19x* and its functional form is implemented in `inc/NangaParbat/PV19x.h`. Other parameterizations are also available, including in particular the functional form chosen for Ref. [84] (in `inc/NangaParbat/PV17.h`).

From the point of view of the code, each parameterization is a different class, derived from the parent class `NangaParbat::Parameterisation`. The "Parameterisation" class is encoded in `inc/NangaParbat/parameterisation.h` and contains some basic functions, implemented in `src/parameterisation/parameterisation.cc`, that, among other functions, set the free parameters of the parameterization and evaluate $f_{\rm NP}$ whenever it is called in the fit.

For every parameterization that can be used in a `NangaParbat` extraction there is a specific configuration card in `cards/`. This `yaml` file contains, besides the general features of the fit, the starting values of the free parameters of the parameterization (this is the reason of why a different configuration file is needed for each functional

form: not all of them have the same parameters or even the same number of free inputs).

For example, this is the configuration file `cards/fitPV19x.yaml` that we used to obtain our nominal result in Ref. [95]:

```
# Short description of the fit that will appear in the report
# of the fit
Description: PV19 version x

# Minimiser to be used for the fit. Possible options so far
# are 'minuit', 'ceres', and 'none'.
Minimiser: minuit

# Seed used with the random-number generator for the
# generation of the Monte Carlo replicas
Seed: '1234'

# Cut on qT / Q. This has to be smaller than the production
# cut used to produce the tables.
qToQmax: '0.2'

# Percentile cut (in percent) on the distribution of chi2's,
# error functions and parameters used to identify outliers.
Percentile cut: '5'

# Enable or disable the t0 prescription for the treatment of
# normalisation uncertantities and define the set
# of parameters to be used to compute the corresponding
# predictions (used only if the t0 prescription is enabled).
# They have to be as many and ordered as the "Parameters"
# below.
t0prescription: true
t0parameters: [0.02351590841248, 0.9422997060112,
   0.2036619541611, 0.326688434032, 0.6715638120608,
   0.04591270083373, 0.06702302432024, 0.4193674275496,
   0.01624238607691]

# Parameterisation to be fitted to data.
# This has to correspond to a derivation of the
# "Parameterisation" class.
Parameterisation: PV19x

# Fluctuate initial parameters according to their step
Paramfluct: false

# List of parameters to be fitted to data. This requires that
   the
```

```
# number and order of parameters matches those expected by the
# particular parameterisation being used.
Parameters:
- {name: g2,      starting_value: 0.02351590841248, step:
   0.006586509, fix: false}
- {name: N1,      starting_value:  0.9422997060112, step:
   0.405145610, fix: false}
- {name: alpha,   starting_value:  0.2036619541611, step:
   0.011498155, fix: false}
- {name: sigma,   starting_value:   0.326688434032, step:
   0.060998937, fix: false}
- {name: lambda,  starting_value:  0.6715638120608, step:
   0.111466055, fix: false}
- {name: N1B,     starting_value: 0.04591270083373, step:
   0.011244147, fix: false}
- {name: alphaB,  starting_value: 0.06702302432024, step:
   0.010590214, fix: false}
- {name: sigmaB,  starting_value:  0.4193674275496, step:
   0.113316907, fix: false}
- {name: g2B,     starting_value: 0.01624238607691, step:
   0.002416170, fix: false}
```

Many of the entries in `cards/fitPV19x.h` have a self-explanatory description already present in the file. In particular, the keys `tOprescription`, `tOparameters` refer to the prescription explained in Sec. 3.6.2, `Seed`, `qToQmax` refer to data points and replica generation (see Sec. 3.3), `Percentile cut` is a parameter needed to produce the report (Sec. 3.2.3), `Description` and `Minimiser` are general features of the fit (Sec. 3.2.1). The entries that are related to the parameterization are: `Parameterisation`, `Paramfluct` and the list `Parameters`.

The starting value of the parameters in the file shown above has been determined through previous fits, as well as the step associated with them, which is the error (relative to each parameter) estimated by `Minuit2`[15].

When `Paramfluct:true`, all the free parameters (`fix: false`) are randomly shifted with a Gaussian noise whose standard deviation is equal to their step. Otherwise, the initial value of each parameter is equal to `starting_value`. The fluctuation of the initial parameters is not to be confused with the fluctuation of the data points of each dataset (see Sec. 3.3): both the procedures are based on random oscillations with Gaussian distribution, but the former is an optional feature of the fit that can help the exploration of the parameter's phase-space, while the latter is the method through which our Monte Carlo TMD replicas are generated.

[15]This assumes a trial–and–error procedure: the values in the parameter list of `cards/fitPV19x.h` are those for which we observed that no further minimization could be obtained.

If no minimization is required when running the fit, the output of `NangaParbat` are predictions obtained with the values `starting_value` of the parameters and the χ^2 is computed accordingly, without any variation of the transverse nonperturbative part.

It is useful to mention here that also the t_0 parameters are determined *a posteriori*: as discussed in Sec. 3.6.2, for a consistent treatment of the uncertainties in the computation of the χ^2, one of the possible choices that can be made is to use the so–called t_0 prescription with parameters equal to theoretical predictions. In this case, the values `t0parameters` in `cards/fitPV19x.h` are outputs of a previous fit, and each t_0 parameter is equal to the starting point of the corresponding $f_{\rm NP}$ parameter.

Finally, it is possible to add user–made parameterizations to the ones already existing in `NangaParbat`. To create a new parameterization for $f_{\rm NP}$, the desired functional form has to be implemented in an header file in `inc/NangaParbat/` following the example of `inc/NangaParbat/PV19x.h`.

Moreover, to make the new $f_{\rm NP}$ appear as a possible choice when running `cli/fit.py`, the file `inc/NangaParbat/nonpertfuncions.h` has to be modified accordingly, inserting the name of the parameterization in:

```
/**
 * @brief Map of currently available parameterisations.
 * Each of them must correspond to a header file
 * containing a class deriving from the
 * NangaParbat::Parameterisation mother class.
 */
const std::map<std::string, Parameterisation*> AvPars
{
  {"DWS",    new NangaParbat::DWS{}},
  {"PV17",   new NangaParbat::PV17{}},
  {"PV19x", new NangaParbat::PV19x{}}
};
```

3.5 Numerical integrations in NangaParbat

The aim of this section is to give a general overview of the numerical implementation in `NangaParbat` of the theoretical formulas in Ch. 2. In particular, we discuss here the explicit formulas encoded in our framework. `NangaParbat` uses `APFEL++` [181, 182] as an engine for the computation of the theoretical predictions.

From Eq. (2.10), the Drell–Yan cross section can be written in terms of TMDs, as

$$\frac{d\sigma}{dQdydq_T} = \frac{16\pi\alpha_{\rm em}^2 q_T}{9Q^3} H(Q,\mu) \sum_q c_q(Q) \int \frac{d^2\mathbf{b}}{4\pi} e^{i\mathbf{b}\cdot\mathbf{q}_T} \overline{F}_q(x_1,\mathbf{b};\mu,\zeta)\overline{F}_{\bar{q}}(x_2,\mathbf{b};\mu,\zeta)\,, \tag{3.7}$$

where we used the short–hand notation

$$\overline{F}_q(x,\mathbf{b};\mu,\zeta) \equiv x\tilde{f}_1^q(x,\mathbf{b};\mu,\zeta)\,, \tag{3.8}$$

As discussed in Ch. 2, the scales μ and ζ are introduced as a consequence of the removal of ultraviolet and rapidity divergences in the definition of the TMDs and are typically set equal to the vector boson's virtuality, $\mu = \sqrt{\zeta} = Q$.

The integral that represents the computationally–intensive part of Eq.(3.7) is:

$$I_{ij}(x_1,x_2,q_T;\mu,\zeta) = \int \frac{d^2\mathbf{b}}{4\pi} e^{i\mathbf{b}\cdot\mathbf{q}_T}\overline{F}_i(x_1,\mathbf{b};\mu,\zeta)\overline{F}_j(x_2,\mathbf{b};\mu,\zeta)\,. \tag{3.9}$$

where $\overline{F}_{i(j)}$ are combinations of evolved TMD PDFs. For convenience, the indexes i and j do not coincide with q and $\bar{q}$ but they are linked through a simple linear transformation (see Eq. (3.14).

As can be seen from Eq. (3.9), in principle a bidimensional integral has to be computed. However, $\overline{F}_{i(j)}$ only depend on the absolute value of $\mathbf{b}$, $|\mathbf{b}| = b_T$, and using Eq (2.9) we can write:

$$I_{ij}(x_1,x_2,q_T;\mu,\zeta) = \frac{1}{2}\int_0^\infty db_T\, b_T J_0(b_T q_T)\overline{F}_i(x_1,b_T;\mu,\zeta)\overline{F}_j(x_2,b_T;\mu,\zeta)\,. \tag{3.10}$$

where J_0 is the zero-th order Bessel function of the first kind. Then, using the b_* prescription introduced in Sec. 2.1.5 and including the nonperturbative function defined in Eq. (2.39), Eq. (3.10) becomes:

$$\begin{aligned} I_{ij}(x_1,x_2,q_T;\mu,\zeta) &= \int_0^\infty db_T\, J_0(b_T q_T)\left[\frac{b_T}{2}\overline{F}_i(x_1,b_*(b_T);\mu,\zeta)\overline{F}_j(x_2,b_*(b_T);\mu,\zeta)\right. \\ &\qquad \left. f_{\rm NP}(x_1,b_T,\zeta)f_{\rm NP}(x_2,b_T,\zeta)\right] \\ &= \frac{1}{q_T}\int_0^\infty d\overline{b_T}\, J_0(\overline{b_T})\left[\frac{\overline{b_T}}{2q_T}\overline{F}_i\left(x_1,b_*\left(\frac{\overline{b_T}}{q_T}\right);\mu,\zeta\right)\overline{F}_j\left(x_2,b_*\left(\frac{\overline{b_T}}{q_T}\right);\mu,\zeta\right)\right. \\ &\qquad \left. f_{\rm NP}\left(x_1,\frac{\overline{b_T}}{q_T},\zeta\right) f_{\rm NP}\left(x_2,\frac{\overline{b_T}}{q_T},\zeta\right)\right] \end{aligned} \tag{3.11}$$

The integral in Eq. (3.11) is a Hankel transform and can be efficiently computed using the so–called Ogata quadrature [180], which is an interpolation–type quadrature formula based on the zeros of the Bessel function J_ν and is particularly suited for

integrations of oscillatory functions over infinite intervals.

Then, the computation of the integral in Eq. (3.9) is achieved through the weighted sum:

$$I_{ij}(x_1,x_2,q_T;\mu,\zeta) \simeq \frac{1}{q_T}\sum_{n=1}^{N}\frac{w_n^{(0)}z_n^{(0)}}{2q_T}\overline{F}_i\left(x_1,b_*\left(\frac{z_n^{(0)}}{q_T}\right);\mu,\zeta\right) \times \overline{F}_j\left(x_2,b_*\left(\frac{z_n^{(0)}}{q_T}\right);\mu,\zeta\right) f_{\rm NP}\left(x_1,\frac{z_n^{(0)}}{q_T},\zeta\right) f_{\rm NP}\left(x_2,\frac{z_n^{(0)}}{q_T},\zeta\right), \tag{3.12}$$

where the unscaled coordinates $z_n^{(0)}$ and the weights $w_n^{(0)}$ can be precomputed in terms of the zeros of the Bessel function J_0 and one single parameter [16] (see Ref. [180] for more details, specifically Eqs. (5.1) and (5.2) or Appendix 3.B for the relevant formula to compute the unscaled coordinates and the weights).

One of the advantages of Eq. (3.12) is that the infinite sum that approximates the integral can be truncated fairly soon, after a small number of function evaluations (see Ref. [180], in particular Sec.5, for the proof of this statement and for more details). Based on the (empirically verified) assumption that the absolute value of each term in the sum in the right hand side of Eq. (3.12) is smaller than that of the preceding one, the truncation number N is chosen dynamically in such a way that the $(N+1)$-th term is smaller in absolute value than a user–defined cutoff relatively to the sum of the preceding N terms. This dynamical truncation procedure is implemented in `src/fastinterface/convolutiontable.cc`, in `ConvolutionTable::Convolute`.

To compute the sum in Eq. (3.12), we employ the expression in Eq. (2.17) for the TMD PDF $\overline{F}_j$, that we reproduce here:

$$\overline{F}_i(x,b;\mu_0,\zeta_0) = \sum_{j=g,q(\bar{q})} x\int_x^1\frac{dy}{y}C_{ij}(y;\mu_0,\zeta_0)f_j\left(\frac{x}{y},\mu_0\right). \tag{3.13}$$

It is useful to mention here that in QCD the most convenient basis for the matching in Eq. (3.13) is the so–called *evolution* basis (*i.e.* Σ, V, T_3, V_3, etc.). In fact, in this basis the operator matrix C_{ij} is almost diagonal with the only exception of crossing terms that couple the gluon and the singlet Σ distributions [183].

Since TMDs in Eq. (3.7) appear in the so–called *physical* basis (*i.e.* d, $\bar{d}$, u, $\bar{u}$, etc.),

[16] The superscript 0 in $z_n^{(0)}$ and $w_n^{(0)}$ indicates that here we are performing an Hankel transform that involves the Bessel function of order zero J_0. This is useful in view of Sec. 3.5.1, in which the integration over q_T gives rise to a similar Hankel transform with J_0 replaced by J_1. The Ogata quadrature algorithm can be applied also in that case, but coordinates and weights will be different.

we need to rotate $F_{i(j)}$ from the evolution basis, over which the indices i and j run, to the physical basis. This is done by means of an appropriate constant matrix T, so that:

$$\overline{F}_q(x_1, b; \mu, \zeta) = \sum_i T_{qi} F_i(x_1, b; \mu, \zeta)\,, \tag{3.14}$$

and similarly for $\overline{F}_{\bar{q}}$[17].

Putting all pieces together, one can conveniently write the cross section in Eq. (3.7) as:

$$\frac{d\sigma}{dQdyd q_T} \simeq \sum_{n=1}^{N} w_n^{(0)} \frac{z_n^{(0)}}{q_T} S\left(x_1, x_2, \frac{z_n^{(0)}}{q_T}; \mu, \zeta\right) f_{\rm NP}\left(x_1, \frac{z_n^{(0)}}{q_T}, \zeta\right) f_{\rm NP}\left(x_2, \frac{z_n^{(0)}}{q_T}, \zeta\right)\,, \tag{3.15}$$

with:

$$S(x_1, x_2, b; \mu, \zeta) = \frac{8\pi\alpha^2}{9Q^3} H(Q, \mu) \sum_q C_q(Q) \left[\overline{F}_q\left(x_1, b_*(b); \mu, \zeta\right)\right] \left[\overline{F}_{\bar{q}}\left(x_2, b_*(b); \mu, \zeta\right)\right]\,. \tag{3.16}$$

With Eq. (3.15) it is possible to precompute the weights S in such a way that the differential cross section in Eq. (3.7) can be calculated as a simple weighted sum of the transverse non-perturbative contribution. The fact that in Eq. (3.16) S has five arguments can be misleading: in fact, such variables are not independent of each other and S actually depends only on three variables. The two arbitrary scales μ and ζ are usually set to be proportional to Q by a constant factor. In addition, x_1 and x_2 depend on Q and y through

$$x_{1,2} = \frac{Q}{\sqrt{s}} e^{\pm y}\,. \tag{3.17}$$

Therefore, the full dependence on the kinematics of the final state of Eq. (3.7) can be specified by Q, y and q_T.

Despite the fact that Eq. (3.15) provides a powerful tool for a fast computation of cross sections, it is often not sufficient to allow for a direct comparison with experimental data. In fact, experimental measurements of differential distributions are usually delivered as integrated over finite regions of the final–state kinematic phase space, and so the basic quantity to be compared to data is:

$$\frac{d\sigma}{dq_T} = \frac{1}{q_{T,\rm max} - q_{T,\rm min}} \int_{y_{\rm min}}^{y_{\rm max}} dy \int_{Q_{\rm min}}^{Q_{\rm max}} dQ \int_{q_{T,\rm min}}^{q_{T,\rm max}} dq_T \left[\frac{d\sigma}{dQdyd q_T}\right]\,, \tag{3.18}$$

where the ranges $[y_{\rm min}\colon y_{\rm max}]$, $[Q_{\rm min}\colon Q_{\rm max}]$, and $[q_{T,\rm min}\colon q_{T,\rm max}]$ define the phase-space

[17] In `NangaParbat`, this transformation is done in `src/fastinterface/fastinterface.cc`, `FastInterface::LuminosityDY`, calling the appropriate function from `APFEL++`.

integration region and the integrand is given in Eq. (3.7). As a consequence, in order to compare theoretical predictions to data, it is necessary to carry out these integrations. These nested integrals, if evaluated numerically, represent a heavy task that makes an extraction of TMDs from Drell-Yan data computationally very intensive and thus slow. While the integrals over Q and y do need to be computed numerically, the integration in q_T can be carried out analytically, and this greatly reduces numerical computations and the amount of time required.

3.5.1 Integrating over q_T

The integration over bins in q_T can be carried out analytically exploiting the following property of Bessel functions:

$$\frac{d}{dx}\left[x^m J_m(x)\right] = x^m J_{m-1}(x)\,, \tag{3.19}$$

This property connects two Bessel functions of consecutive orders, and in particular we can write:

$$\int dx\, x J_0(x) = x J_1(x) \quad \Rightarrow \quad \int_{x_1}^{x_2} dx\, x J_0(x) = x_2 J_1(x_2) - x_1 J_1(x_1)\,. \tag{3.20}$$

Then, considering Eqs. (3.7), (3.9) and (3.10) and neglecting for the moment the dependence on q_T of the phase–space reduction factor $\mathcal{P}$[18](see the following section on kinematic cuts), we observe that the Drell–Yan differential cross section has the following structure:

$$\frac{d\sigma}{dQdyd q_T} \propto \int_0^\infty db_T\, q_T J_0(b_T q_T) S(b_T) \tag{3.21}$$

where S is a function that does not depend on q_T. So we can apply Eq. (3.20) and the integration over the q_T bin can be written as:

$$\begin{aligned}
&\int_{q_{T,\min}}^{q_{T,\max}} dq_T \left[\frac{d\sigma}{dQdyd q_T}\right] \propto \int_0^\infty db_T \int_{q_{T,\min}}^{q_{T,\max}} dq_T\, q_T J_0(b_T q_T) S(b_T) = \\
&= \int_0^\infty \frac{db_T}{b_T^2} S(b_T) \int_{b_T q_{T,\min}}^{b_T q_{T,\max}} dx\, x J_0(x) \\
&= \int_0^\infty \frac{db_T}{b_T} \left[q_{T,\max} J_1(b_T q_{T,\max}) - q_{T,\min} J_1(b_T q_{T,\min})\right] S(b_T)\,.
\end{aligned} \tag{3.22}$$

[18]Neglecting $\mathcal{P}$ dependence on q_T is strictly correct for inclusive observables in the final–state leptons.

where, between the first and the second line, the change of variable $x = b_T q_T$ was needed in order to have the integrand in the form $xJ_0(x)$ and to be able to apply Eq. (3.20). Then, defining $K(Q, y, q_T)$ as the indefinite integral over q_T of the cross section in Eq. (3.7),

$$K(q_T) \equiv \int dq_T \left[\frac{d\sigma}{dQdydq_T}\right] , \tag{3.23}$$

we can write:

$$\int_{q_{T,\min}}^{q_{T,\max}} dq_T \left[\frac{d\sigma}{dQdydq_T}\right] = K(Q, y, q_{T,\max}) - K(Q, y, q_{T,\min}) , \tag{3.24}$$

Looking at Eq. (3.7) and at the last term of Eq. (3.22), we can see that the explicit expression for K is:

$$\begin{aligned} K(Q, y, q_T) = {} & \frac{\cancel{16}^{8}\pi\alpha^2 q_T}{9Q^3} H(Q, \mu) \\ & \times \int_0^\infty \frac{db_T}{\cancel{2}b_T} \cancel{b_T} J_1(b_T q_T) \sum_q c_q(Q) \overline{F}_q(x_1, b; \mu, \zeta) \overline{F}_{\bar{q}}(x_2, b; \mu, \zeta) f_{\rm NP}(x_1, b, \zeta) f_{\rm NP}(x_2, b, \zeta) \end{aligned} \tag{3.25}$$

The integral in Eq. (3.25), then can be computed using the Ogata quadrature as:

$$K(Q, y, q_T) \simeq \sum_{n=1}^{N} w_n^{(1)} S\left(x_1, x_2, \frac{z_n^{(1)}}{q_T}; \mu, \zeta\right) f_{\rm NP}\left(x_1, \frac{z_n^{(1)}}{q_T}, \zeta\right) f_{\rm NP}\left(x_2, \frac{z_n^{(1)}}{q_T}, \zeta\right) , \tag{3.26}$$

with S defined in Eq. (3.16). The unscaled coordinates $z_n^{(1)}$ and the weights $w_n^{(1)}$ can again be precomputed and stored in terms of the zeros of the Bessel function J_1. Eq. (3.24) reduces the integration in q_T to a calculation completely analogous to the unintegrated cross section. This is particularly convenient because it allows us to avoid the computation of a numerical integration.

Kinematic cuts

In the presence of kinematic cuts, such as those on the final–state leptons, the analytic integration over q_T discussed above cannot be directly performed. The reason is that the implementation of these cuts effectively introduces the q_T–dependent function $\mathcal{P}$ in the integral

$$\frac{d\sigma}{dQdydq_T} = \int_0^\infty db_T \, S(b_T) \mathcal{P}(q_T) q_T J_0(b_T q_T) , \tag{3.27}$$

where, as in Eq. (3.21), $S(b_T)$ indicates the remaining terms which do not depend on q_T. The fact that the coefficient that expresses the kinematic cuts on the final–state leptons is q_T–dependent[19] prevents the direct use of Eq. (3.20). However, taking into account the following considerations, it is still possible to achieve a (semi) analytical integration over the q_T bins.

We note that $\mathcal{P}$ is a slowly-varying function of q_T over the typical bin size: this allows one to approximate the integral over the bins in q_T as

$$\begin{aligned}\int_{q_{T,\min}}^{q_{T,\max}} dq_T\, q_T J_0(b_T q_T)\mathcal{P}(q_T) &\simeq \mathcal{P}\left(\frac{q_{T,\max}+q_{T,\min}}{2}\right)\int_{q_{T,\min}}^{q_{T,\max}} dq_T\, q_T J_0(b_T q_T) \\ &= \mathcal{P}\left(\frac{q_{T,\max}+q_{T,\min}}{2}\right)\frac{1}{b_T}\left[q_{T,\max}J_1(b_T q_{T,\max}) - q_{T,\min}J_1(b_T q_{T,\min})\right] .\end{aligned} \tag{3.28}$$

Unfortunately, this structure is inconvenient because it mixes different bin bounds and prevents a recursive computation. However, it is possible to go further and, assuming that the bin width is small enough, $\mathcal{P}$ can be expanded in the following two equivalent ways:

$$\mathcal{P}\left(\frac{q_{T,\max}+q_{T,\min}}{2}\right) = \begin{cases} \mathcal{P}\left(q_{T,\min}+\Delta q_T\right) \simeq \mathcal{P}\left(q_{T,\min}\right) + \mathcal{P}'\left(q_{T,\min}\right)\Delta q_T \\ \mathcal{P}\left(q_{T,\max}-\Delta q_T\right) \simeq \mathcal{P}\left(q_{T,\max}\right) - \mathcal{P}'\left(q_{T,\max}\right)\Delta q_T \end{cases} , \tag{3.29}$$

with

$$\Delta q_T = \frac{q_{T,\max}-q_{T,\min}}{2} . \tag{3.30}$$

Plugging the expansions above into Eq. (3.28), we find

$$\begin{aligned} b_T\int_{q_{T,\min}}^{q_{T,\max}} dq_T\, q_T J_0(b_T q_T)\mathcal{P}(q_T) \;\simeq\; & q_{T,\max}J_1(b_T q_{T,\max})\left[\mathcal{P}\left(q_{T,\max}\right) - \mathcal{P}'\left(q_{T,\max}\right)\Delta q_T\right] \\ - \;& q_{T,\min}J_1(b_T q_{T,\min})\left[\mathcal{P}\left(q_{T,\min}\right) + \mathcal{P}'\left(q_{T,\min}\right)\Delta q_T\right] . \end{aligned} \tag{3.31}$$

The advantage of this formula as compared to Eq. (3.28) is that each of the terms in the right hand side depends on one single bin bound in q_T rather than on a combination of two consecutive bounds. This allows for a recursive computation of predictions in neighboring bins in q_T.

Therefore, in the presence of kinematic cuts, the actual form of the primitive func-

[19] The function $\mathcal{P}$ also depends on the invariant mass Q and on the rapidity y of the lepton pair, so it needs to be integrated also over those variables. Such integrations are discussed in the following section, Sec. 3.5.2.

tion K defined in Eq. (3.24) and given explicitly in Eq. (3.25) is:

$$K(Q,y,q_T) = \frac{8\pi\alpha^2 q_T}{9Q^3} H(Q,\mu)\left[\mathcal{P}\left(Q,y,q_T\right) \pm \mathcal{P}'\left(Q,y,q_T\right)\Delta q_T\right]$$

$$\times \int_0^\infty db\, J_1(bq_T) \sum_q C_q(Q)\overline{F}_q(x_1,b;\mu,\zeta)\overline{F}_{\bar{q}}(x_2,b;\mu,\zeta) f_{\rm NP}(x_1,b,\zeta) f_{\rm NP}(x_2,b,\zeta)\,, \tag{3.32}$$

where we have explicitly reinstated the dependence of the function $\mathcal{P}$ and its derivative with respect to q_T, $\mathcal{P}'$, on Q and y. In the square bracket in Eq. (3.32), the minus sign applies when q_T is the upper bound of the bin and the plus sign when it is the lower bound (see Eq. (3.31)). As discussed in Sec. 3.5.2, when integrating over bins in Q and y, one should also integrate the functions $\mathcal{P}$ and $\mathcal{P}'$. However, it can be shown that, with in the interpolation procedure discussed in Sec. 3.5.2, these functions can be extracted from the integrals in Q and y in a proper manner in such a way to avoid computing the expensive function $\mathcal{P}$ many times. Moreover, the interpolation procedure simplifies enormously the structure of the resulting interpolation tables.

3.5.2 Numerical integrations

As a final step, we need to perform the integrals over Q and y defined in Eq. (3.18). To compute these integrals we can rely only on numerical methods. Having reduced the integration in q_T to the difference of the two terms in the right hand side of Eq. (3.24)([20]), we can focus on integrating the function K over Q and y for a fixed value of q_T:

$$\widetilde{K}(q_T) = \int_{Q_{\rm min}}^{Q_{\rm max}} dQ \int_{y_{\rm min}}^{y_{\rm max}} dy\, K(Q,y,q_T)\,. \tag{3.33}$$

In this way, the cross section

$$\widetilde{\sigma} = \int_{Q_{\rm min}}^{Q_{\rm max}} dQ \int_{y_{\rm min}}^{y_{\rm max}} dy \int_{q_{T,\rm min}}^{q_{T,\rm max}} dq_T \left[\frac{d\sigma}{dQdydq_T}\right] \tag{3.34}$$

can be calculated as

$$\widetilde{\sigma} = \widetilde{K}(q_{T,\rm max}) - \widetilde{K}(q_{T,\rm min})\,. \tag{3.35}$$

To explain how the integrals in Eq. (3.33) are performed, it is convenient to make explicit the dependence of x_1 and x_2 on Q and y using Eq. (3.17). In addition, for sake of simplicity we will identify the scales μ and $\sqrt{\zeta}$ with Q (possible scale variations

[20] For the moment, the complication introduced by the presence of cuts on the lepton final state discussed in Sect. 3.5.1 is omitted. This issue is addressed at the end of this section.

can be easily reinstated at a later stage) and thus drop one of the arguments from the TMD distributions $\overline{F}$ and from the hard factor H. This yields

$$\begin{aligned}\widetilde{K}(q_T) &= \frac{8\pi q_T}{9}\int_0^\infty db_T\, J_1(b_T q_T)\int_{Q_{\min}}^{Q_{\max}} dQ \int_{e^{y_{\min}}}^{e^{y_{\max}}}\frac{d\xi}{\xi} \\ &\times \frac{1}{Q^3}\alpha^2(Q)H(Q)\sum_q C_q(Q)\overline{F}_q\left(\frac{Q}{\sqrt{s}}\xi, b_*(b_T); Q\right)\overline{F}_{\bar{q}}\left(\frac{Q}{\sqrt{s}}\frac{1}{\xi}, b_*(b_T); Q\right) \\ &\times f_{\rm NP}\left(\frac{Q}{\sqrt{s}}\xi, b_T; Q\right) f_{\rm NP}\left(\frac{Q}{\sqrt{s}}\frac{1}{\xi}, b_T; Q\right),\end{aligned} \tag{3.36}$$

where we have performed the change of variable $e^y = \xi$. Now we define one grid in ξ, $\{\xi_\alpha\}$ with $\alpha = 0, \ldots, N_\xi$, and one grid in Q, $\{Q_\tau\}$ with $\tau = 0, \ldots, N_Q$, each of which with a set of interpolating functions $\mathcal{I}$ associated. In addition, the grids are such that: $\xi_0 = e^{y_{\min}}$ and $\xi_{N_\xi} = e^{y_{\max}}$, and $Q_0 = Q_{\min}$ and $Q_{N_Q} = Q_{\max}$. This allows us to interpolate the pair of functions $f_{\rm NP}$ in Eq. (3.36) for generic values of ξ and Q as:

$$\begin{aligned}&f_{\rm NP}\left(\frac{Q}{\sqrt{s}}\xi, b_T; Q\right) f_{\rm NP}\left(\frac{Q}{\sqrt{s}}\frac{1}{\xi}, b_T; Q\right) \simeq \\ &\simeq \sum_{\alpha=0}^{N_\xi}\sum_{\tau=0}^{N_Q}\mathcal{I}_\alpha(\xi)\mathcal{I}_\tau(Q) f_{\rm NP}\left(\frac{Q_\tau}{\sqrt{s}}\xi_\alpha, b_T; Q_\tau\right) f_{\rm NP}\left(\frac{Q_\tau}{\sqrt{s}}\frac{1}{\xi_\alpha}, b_T; Q_\tau\right).\end{aligned} \tag{3.37}$$

Plugging the equation above into Eq. (3.36) we obtain:

$$\begin{aligned}\widetilde{K}(q_T) &\simeq \frac{8\pi q_T}{9}\int_0^\infty db_T\, J_1(b_T q_T)\sum_{\tau=0}^{N_Q}\sum_{\alpha=0}^{N_\xi}\left[\int_{Q_{\min}}^{Q_{\max}} dQ\,\mathcal{I}_\tau(Q)\,\frac{1}{Q^3}\alpha^2(Q)H(Q)\right. \\ &\times \left.\int_{e^{y_{\min}}}^{e^{y_{\max}}} d\xi\,\mathcal{I}_\alpha(\xi)\,\frac{1}{\xi}\sum_q C_q(Q)\overline{F}_q\left(\frac{Q}{\sqrt{s}}\xi, b_*(b_T); Q\right)\overline{F}_{\bar{q}}\left(\frac{Q}{\sqrt{s}}\frac{1}{\xi}, b_*(b_T); Q\right)\right] \\ &\times f_{\rm NP}\left(\frac{Q_\tau}{\sqrt{s}}\xi_\alpha, b_T; Q_\tau\right) f_{\rm NP}\left(\frac{Q_\tau}{\sqrt{s}}\frac{1}{\xi_\alpha}, b_T; Q_\tau\right).\end{aligned} \tag{3.38}$$

Finally, the integration over b_T can be performed using the Ogata quadrature as discussed above, so that:

$$\widetilde{K}(q_T) \simeq \sum_{n=1}^{N}\sum_{\tau=0}^{N_Q}\sum_{\alpha=0}^{N_\xi}\left[\frac{8\pi}{9}w_n^{(1)}\int_{Q_{\min}}^{Q_{\max}} dQ\,\mathcal{I}_\tau(Q)\,\frac{1}{Q^3}\alpha^2(Q)H(Q)\right.$$
$$\left.\times\int_{e^{y_{\min}}}^{e^{y_{\max}}} d\xi\,\mathcal{I}_\alpha(\xi)\,\frac{1}{\xi}\sum_q C_q(Q)\overline{F}_q\left(\frac{Q}{\sqrt{s}}\xi, b_*\left(\frac{z_n}{q_T}\right);Q\right)\overline{F}_{\bar{q}}\left(\frac{Q}{\sqrt{s}}\frac{1}{\xi}, b_*\left(\frac{z_n}{q_T}\right);Q\right)\right]$$
$$\times f_{\rm NP}\left(\frac{Q_\tau}{\sqrt{s}}\xi_\alpha, \frac{z_n}{q_T}; Q_\tau\right) f_{\rm NP}\left(\frac{Q_\tau}{\sqrt{s}}\frac{1}{\xi_\alpha}, \frac{z_n}{q_T}; Q_\tau\right). \tag{3.39}$$

In conclusion, if we define:

$$W_{n\tau\alpha}(q_T) \equiv \; w_n^{(1)}\frac{8\pi}{9}\int_{Q_{\min}}^{Q_{\max}} dQ\,\mathcal{I}_\tau(Q)\,\frac{\alpha^2(Q)}{Q^3}H(Q)$$
$$\times\int_{e^{y_{\min}}}^{e^{y_{\max}}} d\xi \quad \mathcal{I}_\alpha(\xi)\,\tfrac{1}{\xi}\textstyle\sum_q C_q(Q)\overline{F}_q\left(\frac{Q}{\sqrt{s}}\xi, b_*\left(\frac{z_n}{q_T}\right);Q\right)\overline{F}_{\bar{q}}\left(\frac{Q}{\sqrt{s}}\frac{1}{\xi}, b_*\left(\frac{z_n}{q_T}\right);Q\right), \tag{3.40}$$

the quantity $\widetilde{K}(q_T)$ can be computed as:

$$\widetilde{K}(q_T) \simeq \sum_{n=1}^{N}\sum_{\tau=0}^{N_Q}\sum_{\alpha=0}^{N_\xi} W_{n\tau\alpha}(q_T) f_{\rm NP}\left(\frac{Q_\tau}{\sqrt{s}}\xi_\alpha, \frac{z_n}{q_T}; Q_\tau\right) f_{\rm NP}\left(\frac{Q_\tau}{\sqrt{s}}\frac{1}{\xi_\alpha}, \frac{z_n}{q_T}; Q_\tau\right). \tag{3.41}$$

The advantage of Eq. (3.41) is that the weights $W_{n\alpha\tau}$, that clearly depend on q_T but also on the intervals $[Q_{\min} : Q_{\max}]$ and $[y_{\min} : y_{\max}]$, can be precomputed once and for all for each of the experimental points included in a fit and used to determine the function $f_{\rm NP}$. This provides a fast tool for the computation of predictions that makes the extraction of the non-perturbative part of the TMDs much easier.

We address now the issue of how the weights defined in Eq. (3.40) are affected by the presence of cuts as discussed in Sect. 3.5.1. In principle, the function between square brackets in Eq. (3.32) should be inside the integrals in Eq. (3.40) and integrated over the variable Q and $\xi = e^y$. However, this turns out to be numerically problematic because the phase–space–reduction function $\mathcal{P}$ is expensive to compute. On top of this, the fact that the factor between square brackets in Eq. (3.32) depends on whether q_T is a lower or an upper integration bound would lead to a duplication of the weights to compute.

In order to simplify the computation, we assume that the function $\mathcal{P}$ and its derivative $\mathcal{P}'$ are slowly varying functions of Q and y over the typical grid interval of the

grids in Q and ξ. In addition, the interpolating functions $\mathcal{I}_\tau(Q)$ and $\mathcal{I}_\alpha(\xi)$ are strongly peaked at Q_τ and ξ_α, respectively. These considerations allow us to avoid integrating explicitly $\mathcal{P}$ and $\mathcal{P}'$ over Q and ξ and to replace the weights in Eq. (3.40) with

$$W_{n\tau\alpha}(q_T) \to \left[\mathcal{P}\left(Q_\tau, \ln(\xi_\alpha), q_T\right) \pm \mathcal{P}'\left(Q_\tau, \ln(\xi_\alpha), q_T\right) \Delta q_T\right] W_{n\tau\alpha}(q_T)\,. \tag{3.42}$$

In conclusion, the only additional information required to implement cuts on the final state is the value of the phase–space reduction function $\mathcal{P}$ and its derivative $\mathcal{P}'$ on all points of the bidimensional grid in Q and ξ for all q_T bin bounds: Eq. (3.42) then allows one to use the weights computed over the full phase–space.

Cross section differential in x_F

In some cases, the Drell-Yan differential cross section may be presented as differential in the invariant mass of the lepton pair Q and, instead of the rapidity y, of the Feynman variable x_F defined as

$$x_F = \frac{Q}{\sqrt{s}}\left(e^y - e^{-y}\right) = \frac{2Q}{\sqrt{s}}\sinh y = x_1 - x_2\,. \tag{3.43}$$

This is the case of data of the E605 experiment for low energy Drell–Yan, for example. From the definition of x_F we have:

$$\frac{dx_F}{dy} = \frac{2Q}{\sqrt{s}}\cosh y = x_1 + x_2\,. \tag{3.44}$$

Therefore:

$$\frac{d\sigma}{dQdx_Fdq_T} = \frac{dy}{dx_F}\frac{d\sigma}{dQdydq_T} = \frac{\sqrt{s}}{2Q\cosh y}\frac{d\sigma}{dQdydq_T} = \frac{1}{x_1 + x_2}\frac{d\sigma}{dQdydq_T} \tag{3.45}$$

with:

$$y(x_F, Q) = \sinh^{-1}\left(\frac{x_F\sqrt{s}}{2Q}\right) = \ln\left[\frac{\sqrt{s}}{2Q}\left(x_F + \sqrt{x_F^2 + \frac{4Q^2}{s}}\right)\right]\,, \tag{3.46}$$

so that:

$$x_1 = \frac{1}{2}\left(x_F + \sqrt{x_F^2 + \frac{4Q^2}{s}}\right) \quad \text{and} \quad x_2 = \frac{Q^2}{sx_1}\,. \tag{3.47}$$

Therefore, we can compute the integral:

$$\widetilde{I}(q_T) = \int_{Q_{\min}}^{Q_{\max}} dQ \int_{x_{F,\min}}^{x_{F,\max}} dx_F\, I(Q, x_F, q_T)\,, \tag{3.48}$$

where I is the primitive in q_T of the cross section differential in x_F:

$$I(Q, x_F, q_T) = \int dq_T \left[\frac{d\sigma}{dQ dx_F dq_T}\right] , \tag{3.49}$$

following the same steps of Sect. 3.5.2, for the (numerical) integration over Q and y. This leads to:

$$\widetilde{I}(q_T) \simeq \sum_{n=1}^{N}\sum_{\tau=0}^{N_Q}\sum_{\alpha=0}^{N_x} \overline{W}_{n\tau\alpha}(q_T) f_{\rm NP}\left(x_{1,\alpha\tau}, \frac{z_n}{q_T}; Q_\tau\right) f_{\rm NP}\left(x_{2,\alpha\tau}, \frac{z_n}{q_T}; Q_\tau\right) , \tag{3.50}$$

with:

$$\begin{aligned}\overline{W}_{n\tau\alpha}(q_T) \equiv\ & w_n^{(1)} \frac{8\pi}{9} \int_{Q_{\rm min}}^{Q_{\rm max}} dQ\, \mathcal{I}_\tau(Q)\, \frac{1}{Q^3} \alpha^2(Q) H(Q) \\ & \times \int_{x_{F,\rm min}}^{x_{F,\rm max}} dx_F\, \mathcal{I}_\alpha(x_F)\, \tfrac{1}{x_1+x_2} \textstyle\sum_q C_q(Q) \overline{F}_q\left(x_1, b_*\left(\tfrac{z_n}{q_T}\right); Q\right) \overline{F}_{\bar{q}}\left(x_2, b_*\left(\tfrac{z_n}{q_T}\right); Q\right) ,\end{aligned} \tag{3.51}$$

where x_1 and x_2 are functions of x_F and Q through Eq. (3.47). In addition, we have defined a grid in x_F, $\{x_{F,\alpha}\}$ with $\alpha = 0, \ldots, N_x$, that allowed us to define $x_{1(2),\alpha\tau} \equiv x_{1(2)}(x_{F,\alpha}, Q_\tau)$.

3.6 Computation of χ^2

The figure of merit used to determine the quality of the agreement between experimental data and theoretical predictions is the χ^2 value. This object is also the quantity that gets minimized during a fitting procedure, and it is therefore of the utmost importance to be able to implement its computation efficiently.

The χ^2 definition, for a certain dataset with n data points, is:

$$\chi^2 = \sum_{i,j=1}^{n} (m_i - t_i)\, V_{ij}^{-1}\, (m_j - t_j) , \tag{3.52}$$

where $\{t_i\}$ is a set of theoretical predictions for the given experiment, $\{m_i\}$ are the measurements central values, and V_{ij} is the covariance matrix. Different kinds of uncertainties can be distinguished based on their experimental origin. In general, we can write:

$$m_i \pm \sigma_{i,\rm unc} \pm \sigma_{i,\rm corr}^{(1)} \pm \cdots \pm \sigma_{i,\rm corr}^{(k)} , \tag{3.53}$$

where $\sigma_{i,\rm unc}$ are the uncorrelated uncertainties and $\sigma_{i,\rm corr}^{(1)} \pm \cdots \pm \sigma_{i,\rm corr}^{(k)}$ are correlated

uncertainties associated with the central value m_i.

In case there is more than one uncorrelated statistical uncertainty, $\sigma_{i,\text{unc}}$ is the square root of the sum in quadrature of all the uncorrelated systematic uncertainties associated with the data point m_i.

Also, Eq. (3.53) assumes that the systematic errors are all symmetric: since this is not always the case, we should note that, in order to use the notation above, the systematic errors provided by the experiments have to be symmetrized and the measured central values ought to be consistently shifted according to Eqs.(27)–(28) of Ref. [184].

When the information on the nature of the uncertainties is available, the covariance matrix, that enters in Eq. (3.52), can be computed, and it may be written as

$$V_{ij} = \left(\sigma^2_{i,\text{stat}} + \sigma^2_{i,\text{unc}}\right)\delta_{ij} + \sum_{l=1}^{k} \sigma^{(l)}_{i,\text{corr}}\sigma^{(l)}_{j,\text{corr}} \,. \tag{3.54}$$

Further details about this formula for $\mathbf{V}$ are discussed in Sec. 3.6.2.

3.6.1 Effective χ^2 computation

According to Eq. (3.52), in order to compute the χ^2 value, one has to invert the covariance matrix. This operation is computationally expensive to perform, in terms of time required and numerical calculations, since $\mathbf{V}$ has several rows and columns (one for each data point). Therefore, implementing the inversion of the covariance matrix of Eq. (3.52) with numerical methods is not an optimal choice in a fitting framework.

An alternative method to implement the computation of the inversion of the experimental covariance matrix $\mathbf{V}$ required to evaluate the χ^2 relies on the so–called Cholesky decomposition. This is an iterative procedure that exploits the fact that $\mathbf{V}$ is a symmetric matrix, as can be seen from Eq. 3.54.

It can be proven that any symmetric and positive definite matrix $\mathbf{V}$ can be decomposed in terms of a lower triangular matrix $\mathbf{L}$ as

$$\mathbf{V} = \mathbf{L} \cdot \mathbf{L}^T \,, \tag{3.55}$$

The entries of $\mathbf{L}$ are obtained recursively from those of the covariance matrix as follows:

$$
\begin{aligned}
L_{kk} &= \sqrt{V_{kk} - \sum_{j=1}^{k-1} L_{kj}^2}\,, \\
L_{ik} &= \frac{1}{L_{kk}} \left(V_{ik} - \sum_{j=1}^{k-1} L_{ij} L_{kj} \right)\,, \quad k < i\,, \\
L_{ik} &= 0\,, \quad k > i\,.
\end{aligned}
\tag{3.56}
$$

Writing the χ^2 definition 3.52 in the matrix, it can be clearly seen that, defining $y_i \equiv m_i - t_i$, we have:

$$
\chi^2 = \mathbf{y}^T \cdot \mathbf{V}^{-1} \cdot \mathbf{y} = \left| \mathbf{L}^{-1} \cdot \mathbf{y} \right|^2 \,. \tag{3.57}
$$

We notice that the vector $\mathbf{x} \equiv \mathbf{L}^{-1} \cdot \mathbf{y}$ is the solution of the linear system:

$$
\mathbf{L} \cdot \mathbf{x} = \mathbf{y}\,, \tag{3.58}
$$

where $\mathbf{L}$ and $\mathbf{y}$ are known, so that $\mathbf{x}$ can efficiently be found by forward substitution.[21] Then, the χ^2 may also be written as

$$
\chi^2 = |\mathbf{x}|^2 \,. \tag{3.59}
$$

Therefore, with the Cholesky decomposition, there is no need to perform explicitly the inversion of the covariance matrix $\mathbf{V}$. This simplifies significantly the computation of the χ^2. The procedure described above is implemented in `NangaParbat` in `src/chi2/chisquare.cc` and `src/chi2/linearsystems.cc`.

3.6.2 Additive and multiplicative uncertainties

To have a faithful treatment of the correlated uncertainties in the computation of the χ^2, particular attention has to be payed to the distinction between *additive* and *multiplicative* uncertainties. This is due to the fact that they play a different role in the entries of the covariance matrix, and, specifically, including multiplicative uncertainties in the wrong way can result in the so-called D'Agostini bias [185].

The nature of each uncertainty is typically provided by experimental collaborations in their official releases of the datasets. A multiplicative uncertainty is an uncertainty

[21] Forward substitution is a modification of the general substitution method and is the procedure that solves a linear system $\mathbf{L} \cdot \mathbf{x} = \mathbf{y}$ with a lower triangular coefficient matrix $\mathbf{L}$. Its algorithm can be implemented in a code by simple steps: 1) $x_i = y_i$ 2) $x_i = x_i - \sum_{j<i} L_{ij} x_j$ 3) $x_i = x_i / L_{ii}$.

that affects the normalization of the data. An example is the luminosity uncertainty.

Now, let us express all the correlated systematic uncertainties $\sigma^{(l)}_{i,\text{corr}}$ as relative to their central value m_i, so that we define[22]

$$\sigma^{(l)}_{i,\text{corr}} \equiv \delta^{(l)}_{i,\text{corr}} m_i \,, \tag{3.60}$$

and let us also define $s_i^2 \equiv \sigma^2_{i,\text{stat}} + \sigma^2_{i,\text{unc}}$ so that Eq. (3.54) can be rewritten as

$$V_{ij} = s_i^2 \delta_{ij} + \left(\sum_{l=1}^{k} \delta^{(l)}_{i,\text{corr}} \delta^{(l)}_{j,\text{corr}} \right) m_i m_j \,. \tag{3.61}$$

Then, we split the correlated systematic uncertainties into k_a additive uncertainties and k_m multiplicative uncertainties, such that $k_a + k_m = k$. This way Eq. (3.61) takes the form:

$$V_{ij} = s_i^2 \delta_{ij} + \left(\sum_{l=1}^{k_a} \delta^{(l)}_{i,\text{add}} \delta^{(l)}_{j,\text{add}} + \sum_{l=1}^{k_m} \delta^{(l)}_{i,\text{mult}} \delta^{(l)}_{j,\text{mult}} \right) m_i m_j \,. \tag{3.62}$$

This definition of the covariance matrix, though, is problematic: if Eq. 3.62 was to be used in a fitting procedure, it would induce a bias. Such bias is called the D'Agostini bias, after Ref. [185, 186], where it was studied, and it is a downward shift of the fitted theoretical predictions with respect to the data.

The bias concerns specifically datasets with large normalizations uncertainties and causes predictions that largely undershoot the data. As a consequence, such shift influences indirectly the quality of the whole fit. The direct effect of the bias is much milder in case N experiments are considered.

The magnitude of the downward shift depends on the differences $m_i - m_j$, on the size of the normalization error and on the number of data points considered.

In order to comprehend the origin of the D'Agostini bias, it is useful to consider a simple example. Let $m_1 \pm s_1$ and $m_2 \pm s_2$ be two measurements of the same physical quantity t, [23], from a single experiment with uncorrelated uncertainties s_1 and s_2. Suppose that they only have one correlated uncertainty, a common normalization error $\delta_{1,mult} = \delta_{2,mult} \equiv \delta$.

[22] Note that this redefinition does not change the nature of the uncertainties, additive uncertainties remain additive as well as multiplicative uncertainties remain multiplicative.

[23] For simplicity, here we suppose that we have only one quantity t to be determined. In TMD extractions (as well as in PDFs fits) the situation is more complicated, as t is some nontrivial function (*i.e.*, the cross section) of many fitted parameters that describe the distribution of interest. This further complication does not concern the issue at hand, and considering only one t for illustration purposes takes nothing away from the nature of the bias.

In this case, the covariance matrix calculated with Eq. (3.62) is

$$\mathbf{V} = \begin{pmatrix} s_1^2 + \delta^2 m_1 & \delta^2 m_1 m_2 \\ \delta^2 m_1 m_2 & s_2^2 + \delta^2 m_2 \end{pmatrix} . \tag{3.63}$$

With this covariance matrix, using Eq. (3.52), we can calculate $\chi^2(t)$. After minimizing $\chi^2(t)$ with respect to t, we obtain that the best value of t is

$$t = \frac{s_1^2 m_2 + s_2^2 m_1}{s_1^2 + s_2^2 + (m_1 - m_2)^2 \delta^2} = \frac{w}{1 + \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}\delta^2} . \tag{3.64}$$

With respect to the weighted average $w = (s_1^2 m_2 + s_2^2 m_1)/(s_1^2 + s_2^2)$, Eq. (3.64) has a new term in the denominator, $(m_1 - m_2)^2 \delta^2$. If this term is negligible with respect to the squared sum of the uncorrelated and statistic uncertainties, we get a value very close to the weighted average of the two measurement, otherwise a smaller theoretical prediction is obtained. In other words, as soon as there is a disagreement between the data points, if their normalization error is different from zero, the result for t tends to be shifted to small values.

It can be shown [185, 187] that the bias gets worse as the number of data points gets bigger. For n measurements the theoretical prediction is

$$t = \frac{w}{1 + \sum_{i=1}^{n} \frac{(m_i - w)^2}{s_i^2}\delta^2} , \tag{3.65}$$

where w is the weighted average.

Therefore, in presence of large overall uncertainties on the scale, in order to minimize the χ^2 the fitting procedure tends to underestimate the absolute value of the predictions, preferring normalization factors smaller than 1 (within the range allowed by the relative normalization error).

Ultimately, what leads to incorrect predictions when performing a fit using Eq. (3.62) is the presence, in the covariance matrix, of the term $\sum_{l=1}^{k_m} \delta_{i,\text{mult}}^{(l)} \delta_{j,\text{mult}}^{(l)} m_i m_j$, and specifically the multiplication for the central values of the measurements $m_i m_j$.

Since the bias is exacerbated when the differences between the data points are sizable, or when data are inconsistent with each other, a possible solution to the problem is to use theoretical predictions instead of experimental values when treating multiplicative uncertainties. In fact, theoretical predictions are, by construction, a more precise estimator of the observable than m_i, as they level out the discrepancies in the central value of the different data points.

This is the so–called t_0 prescription [187], where the experimental central value m_i in the multiplicative term is replaced by a fixed theoretical predictions $t_i^{(0)}$.

Applying the t_0 prescription, the covariance matrix takes the form:

$$V_{ij} = s_i^2 \delta_{ij} + \sum_{l=1}^{k_a} \delta_{i,\text{add}}^{(l)} \delta_{j,\text{add}}^{(l)} m_i m_j + \sum_{l=1}^{k_m} \delta_{i,\text{mult}}^{(l)} \delta_{j,\text{mult}}^{(l)} t_i^{(0)} t_j^{(0)} \,. \tag{3.66}$$

The value of t_0 must be fixed beforehand: it has to be tuned to be consistent with the theoretical prediction t resulting from the fit, but there is no standard procedure to determine it in a self–consistent way during the minimization.

In our analysis, as coefficients $t_i^{(0)}$ (one for each parameter resulting from the fit) we use the values of the parameters computed in a previous fit. [24]

The implementation in `NangaParbat` of Eq. (3.66) was essential to take care of the large normalization uncertainties of the dataset from the fixed–target Drell–Yan experiments.

The definition in Eq. (3.66) of the covariance matrix gives a χ^2 function that can be used to produce unbiased fits.

3.6.3 Systematic shifts

In order to visualize the effect of systematic uncertainties, it is instructive to compute the *systematic shift* generated by the systematic uncertainties. To do so, we need to write the χ^2 in terms of the so-called "nuisance parameters" λ_α:

$$\chi^2 = \sum_{i=1}^{n} \frac{1}{s_i^2} \left(m_i - t_i - \sum_{\alpha=1}^{k} \lambda_\alpha \sigma_{i,\text{corr}}^{(\alpha)} \right)^2 + \sum_{\alpha=1}^{k} \lambda_\alpha^2 \,. \tag{3.67}$$

This is an alternative definition of the χ^2, and it can be demonstrated that Eq. (3.67) is equivalent to Eq. (3.52) [188].

An important characteristic of the nuisance parameters is that their optimal value is determined minimizing the χ^2 with respect to them. Since the (3.67) dependence on λ_α is quadratic, the best value for the nuisance parameters can be found analytically by imposing:

$$\frac{\partial \chi^2}{\partial \lambda_\beta} = 0 \,. \tag{3.68}$$

Solving Eq. (3.68) means finding the solution to the system:

$$\sum_{\beta=1}^{k} A_{\alpha\beta} \lambda_\beta = \rho_\alpha \,, \tag{3.69}$$

[24] If the value chosen for t_0 is not consistent with the datasets, this may lead to an incorrect fit; however, the dependence on t_0 of the χ^2 is rather weak, as is shown in Ref. [187]

where:

$$A_{\alpha\beta} = \delta_{\alpha\beta} + \sum_{i=1}^{n} \frac{\sigma_{i,\text{corr}}^{(\alpha)} \sigma_{i,\text{corr}}^{(\beta)}}{s_i^2} \quad \text{and} \quad \rho_\alpha = \sum_{i=1}^{n} \frac{m_i - t_i}{s_i^2} \sigma_{i,\text{corr}}^{(\alpha)} \,. \tag{3.70}$$

In this way, starting from the correlated uncertainties of the experimental datasets, we determine the values of λ_β. Every source of uncertainty has its nuisance parameter λ_β which follows the standard normal distribution.

Now, looking at Eq. (3.67), we find it is useful to define two quantities:

$$d_i = \sum_{\alpha=1}^{k} \lambda_\alpha \sigma_{i,\text{corr}}^{(\alpha)} \tag{3.71}$$

that can be interpreted as a shift induced by the correlated systematic uncertainties; and the shifted predictions $\bar{t}$:

$$\bar{t}_i = t_i + d_i \,, \tag{3.72}$$

With these elements at hand, the χ^2 may be rewritten as:

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{m_i - \bar{t}_i}{s_i} \right)^2 + \sum_{\alpha=1}^{k} \lambda_\alpha^2 \,. \tag{3.73}$$

This expression for the χ^2 contains explicit information about the value of the nuisance parameters (at the best fit).

It can be noticed that Eq. (3.73) is similar to the form of the definition of the χ^2 where only uncorrelated uncertainties are present: with respect to the latter, in (3.73) there is an extra term, called *penalty term*, given by the sum of the nuisance parameters.

Names in this case are particularly evocative: the penalty term, in fact, is the price to pay to recover the uncorrelated form of the χ^2 formula, and the nuisance parameters have to be small. As a rule of thumb, the penalty term is expected to be of modest size, and, although there is no specific rule that says that it has to be smaller than the uncorrelated contribution, if the nuisance parameters dominate the χ^2, this may be an indication of a poor fit.

In `NangaParbat`, (in `src/chi2/chisquare.cc`), both (3.66) and (3.73) are implemented, and during the fitting procedure the numerical equivalence between the χ^2 computed in terms of the nuisance parameters and that computed using the covariance matrix is carefully checked by the code.

Shifted theoretical predictions are an optimal tool to ascertain the agreement between theoretical predictions and data. For this reason, in Sec 4.3, we chose to show in the plots the comparison between central experimental measures m_i and shifted

predictions $\bar{t}_i$ in units of the uncorrelated uncertainty s_i.

3.7 Minimization

In order to fit the function $f_{\rm NP}$ of Eq. (3.15) to data, we used two independent codes: `Minuit2` [189] as implemented in `ROOT`, and `Ceres Solver` [190] (from now on simply `Ceres` in this work). While the first (`Minuit`) has been used for this kind of tasks for many years, the second (`Ceres`) is relatively new and typically used for more complex problems such as image recognition, 3D modeling, etc...

Recently, the `xFitter` Collaboration [191] has used `Ceres Solver` for fitting collinear PDFs [192], showing that this tool is suitable also for this kind of tasks. Having two independent codes within the same framework turned out to be particularly useful to cross check our results: to obtain the nominal results for PV19 we tried both minimizing procedures and made sure that we obtained the same values for the final parameters of $f_{\rm NP}$. While the fact that two minimizers based on different techniques give us the same results does not guarantee that the minimization was successful in finding the absolute minimum for the functional form of $f_{\rm NP}$, it does demonstrate a certain stability of our results.

3.7.1 Minimizers and implementation

When running a fit in `NangaParbat`, the choice of the minimizer is read from the input card selected in `cards/` (see Sec. 3.4.2). The interfaces between `NangaParbat` and `Minuit2` and `Ceres` are implemented in `src/minimisation/` and then called in `run/RunFit.cc`, as shown in the snippet of code below:

```
// Allocate "Parameterisation" derived object
NangaParbat::Parameterisation *NPFunc = NangaParbat::
   GetParametersation(fitconfig["Parameterisation"].as<std::
   string>());

// Define "ChiSquare" object with a given qT / Q cut
NangaParbat::ChiSquare chi2{*NPFunc};

// Set parameters for the t0 predictions using "t0parameters"
// in the configuration card only if the the t0 has been
// enabled and the central replica is not being computed.
if (fitconfig["t0prescription"].as<bool>())
  NPFunc->SetParameters(fitconfig["t0parameters"].as<std::
     vector<double>>());

// Open datasets.yaml file that contains the list of datasets
```

```
14 // to be fitted and push the corresponding pairs of
15 // "DataHandler" and "ConvolutionTable" objects into the
16 // a vector.
17 const YAML::Node datasets = YAML::LoadFile(std::string(argv
       [3]) + "/datasets.yaml");
18 for (auto const& exp : datasets)
19 for (auto const& ds : exp.second)
20   {
21     std::cout << "Reading table for " << ds["name"].as<std::
         string>() << "..." << std::endl;
22
23     // Convolution table
24     const std::string table = std::string(argv[4]) + "/" + ds[
         "name"].as<std::string>() + ".yaml";
25     const NangaParbat::ConvolutionTable ct{YAML::LoadFile(
         table), fitconfig["qToQmax"].as<double>()};
26     //ct.NumericalAccuracy(NPFunc->Function());
27
28     // Datafile
29     const std::string datafile = std::string(argv[3]) + "/" +
         exp.first.as<std::string>() + "/" + ds["file"].as<std::
         string>();
30     const NangaParbat::DataHandler dh{ds["name"].as<std::
         string>(), YAML::LoadFile(datafile), rng, ReplicaID,
31                                       (fitconfig["
                                          t0prescription"].as<
                                          bool>() ? ct.
                                          GetPredictions(NPFunc
                                          ->Function()) : std::
                                          vector<double>{})};
32
33     // Add chi2 block
34     chi2.AddBlock(std::make_pair(dh, ct));
35   }
36 // Report time elapsed
37 t.stop();
38
39 // Minimise the chi2 using the minimiser indicated in the
       input card
40 t.start();
41 bool status;
42 if (fitconfig["Minimiser"].as<std::string>() == "none")
43   status = NoMinimiser(chi2, fitconfig["Parameters"]);
44 else if (fitconfig["Minimiser"].as<std::string>() == "minuit")
45   status = MinuitMinimiser(chi2, fitconfig["Parameters"], (
       fitconfig["Paramfluct"].as<bool>() ? rng : NULL));
46 else if (fitconfig["Minimiser"].as<std::string>() == "ceres")
```

```
47    status = CeresMinimiser(chi2, fitconfig["Parameters"], (
          fitconfig["Paramfluct"].as<bool>() ? rng : NULL));
48  else if (fitconfig["Minimiser"].as<std::string>() == "scan")
49    status = MinuitScan(chi2, fitconfig["Parameters"], std::
          string(argv[1]));
50  else
51    throw std::runtime_error("[RunFit]: Unknown minimiser");
52
53  // Print the total chi2 on screen
54  std::cout << "Total chi2 = " << chi2() << "\n" << std::endl;
55
56  // Produce the output
57  YAML::Emitter out;
58  out << chi2;
59  std::ofstream rout(OutputFolder + "/Report.yaml");
60  rout << "Status: " << status << std::endl;
61  rout << out.c_str() << std::endl;
62  rout.close();
```

As this is where the fit effectively takes place, it is worth to briefly comment the structure of the code above, that in a certain sense reflects the structure of `NangaParbat` and allows a better comprehension of the entire fitting framework.

The core of `run/RunFit.cc` is a *ChiSquare* class object of the, `NangaParbat::ChiSquare chi2`, that needs the parameterization of the nonperturbative function as an input. In order to compute the χ^2, `NangaParbat::ChiSquare chi2` needs also a pair consisting in an object of the *ConvolutionTable* class (perturbative input, see also Sec. 3.4.1) and one of the *DataHandler* class (for more details see Sec. 3.3.2): for each experiment, they are initialized and then added to `chi2` through the `AddBlock` method.

Then, using `chi2`, the selected minimization procedure is called. There are four options available: two of them are actual minimizations, with `Minuit2` or with `Ceres`, `"none"` corresponds to the choice of computing the χ^2 without minimization [25] while with `scan` `NangaParbat` computes the χ^2 without minimization *and* performs also a scanning of the phase–space of each parameter.

The `scan` option is particularly useful in the early stages of the fit, to see if each parameter is reasonably near its minimium or if a further minimization is possible with more iterations, changing the initial parameters in `cards/` accordingly. In fact, sometimes the minimizer reaches a minimum (determined considering some internal parameters for the accuracy) but a further minimization is still possible. In these cases, it is customary to take the output values of the parameters and feed them to

[25] In this case, `Minuit2` is called anyway, but no minimization is performed and the final parameters are put equal to the initial ones. The implementation of `bool NoMinimiser(ChiSquare const& chi2, YAML::Node const& parameters)` can be found in `src/minimisation/minimisation.cc`.

the fit as a new starting point of the minimization. Such operation is done replacing the parameters in the input card (either manually or using the corresponding option in `cli/fit.py`). The parameter scan is implemented exploiting the function `Scan` available in `Minuit2`. If such option is chosen when running the fit with `cli/fit.py` a plot for each parameter is also produced.

The object `chi2` also takes care of producing the output once the fitting procedure is finished. Besides printing some salient characteristics of the fit on the terminal, `NangaParbat` stores the output for each replica in a `yaml` file called `Report.yaml` and placed in `NameOfOutputFolder/replica_N/` (where N is the replica identification number). In particular, the first line of `Report.yaml` indicates if the minimization procedure for the relative replica was successful or not (`Status: 1` means that the replica converged).

The output of `NangaParbat` in Fig. 3.1, with its directories and sub–directories, is the baseline to create a proper report of the fit using `cli/report.py`.

As final remark, we mention that besides the final value of the parameters `Minuit2` also gives an estimate on their errors. However, the uncertainties on the final parameters of our $f_{\rm NP}$ extraction that we discuss in Ch. 4 do not correspond to with the `Minuit2` computed errors, as in our statistical analysis the estimate of a parameter uncertainty is related to the spread of the Monte Carlo replicas around the central value (which is the final value of the parameter). As mentioned in Sec. 3.4.2, the errors estimated by `Minuit` are used in the input card to set the initial step for the eventual fluctuation of the initial values of the parameters.

3.8 Grid production

The suite of tools `NangaParbat` can be also used to produce grids in which results of fits are encoded. Grids are particularly useful to make TMD extractions available to the community. To access the information contained in them, `NangaParbat` also provides interpolation tools, discussed in Sec. 3.9.

We produced grids for both TMD extractions currently implemented in `NangaParbat`, PV17 [84] and PV19 (the extraction discussed in Ch. 4, and published in Ref. [95]). In particular, the following sets of grids are available:

- PV19 unpolarized TMD PDFs, f_1
- PV17 unpolarized TMD PDFs, f_1
- PV17 unpolarized TMD FFs, for $D_1^{h\to\pi^+}$
- PV17 TMD FFs, $D_1^{h\to\pi^-}$

- PV17 structure function $F_{UU,T}^{h\to\pi^+}$
- PV17 structure function $F_{UU,T}^{h\to\pi^-}$

In order to have the possibility of producing a grid for a specific parameterization, such parameterization has to be implemented in `NangaParbat`, which means that its analytical form has to be encoded in a file in `inc/NangaParbat/` and a directory with the replicas' results with the style of a `NangaParbat` output (see Fig.3.1) has to exist.

Our grids are in `yaml` format and are optimized to have a manageable size: in fact, one of the main problems when tabulating functions in three or four dimensions is that the final size of the files can be huge and thus not suitable for an easy transfer and usage.[26]

`NangaParbat` grids in principle can be used with every interpolator that is capable of reading tables structured as is specified in this section. However we suggest the use of the interpolator included in `NangaParbat`. In fact, the philosophy behind grid production in `NangaParbat` is to have grids as light as possible, with points coarsely span the spectrum of each variable, and to provide a good interpolator that allows the user to get sufficiently good results from such grids.

The grids we provide have been thoroughly tested using the `NangaParbat` interpolator, comparing results obtained from grids with direct calculations of the TMDs (see Sec. 3.9.1). Results obtained using other interpolators may significantly differ from predictions obtained with direct calculation and may depend on the choice of interpolator.

Grids produced by the Pavia group are presently available on TMDlib and TMDplotter [193, 194]. In particular, they can be downloaded from:

`https://syncandshare.desy.de/index.php/s/GjjcwKQC93M979e?path=%2FTMD%20grid%20files`

and they can be plotted though the TMDplotter web interface at:

`http://tmdplotter.desy.de/TMDplotter.php?mode=default&type=tmdkt`.

In the following sections, the implementation and the structure of our grids are illustrated.

3.8.1 TMD grids

As mentioned in the introduction, once a TMD fit has been performed with `NangaParbat`, starting from the output it is possible to encode the results of such extraction in three–dimensional grids.

[26] When considering the size of grids relative to a given extraction, one should also consider the number of Monte Carlo replicas available for such extraction: for example, there are 200 replicas both in PV17 and PV19, so the final size of the grid folder is equal to around 200 times the size of a single grid.

In accordance with `NangaParbat`'s structure, the program to launch to produce TMD grids is `run/CreateGrids`, which takes the fit output folder as an input from the terminal line and calls the function `NangaParbat::ProduceTMDGrid`. The second input that `run/CreateGrids` takes is whether to produce TMD PDF grids or TMD FF grids: in case of PV19, only TMD PDF can be considered, but for PV17 the user can choose between those two options.

The functions `NangaParbat::ProduceTMDGrid` and `NangaParbat::EmitTMDGrid` (the second one is called by the first) are responsible for the computation of the grid and its production in `yaml` format. Such functions are implemented in `src/tmdgrid/createtmdgrid.cc`, while in `inc/NangaParbat/createtmdgrid.h` the `NangaParbat` structure dedicated to three–dimensional grids, `struct ThreeDGrid`, is encoded.

The output of `run/CreateGrids` is a folder, whose name is chosen by the user (it is the third element in the terminal input line, without it the program does not run). The folder contains a grid for each replica plus an info file, with the same name of the grid, where the most important features of the extraction and the grid are written As reference, we show the info files for the PV19 TMD PDF extraction and for PV17 FF for hadron in π^+. For PV19 f_1 we have:

```
SetDesc: Set produced with NangaParbat + APFEL++ (please cite
   arXiv:1912.07550 and arXiv:1708.00911)
Authors: A. Bacchetta, V. Bertone, C. Bissolotti, G. Bozzi, F.
    Delcarro, F. Piacenza, M. Radici
Reference: arXiv:1912.07550
SetIndex: 000000
SetName: PV17nll
TMDType: pdf
CollDist: MMHT2014nnlo68cl
CollDistMember: 0
Format: TMDlib2
DataVersion: 1
OrderQCD: NNNLL
AlphaS_OrderQCD: 2
Regularisation: bstarmin
NumMembers: 216
ErrorType: Monte Carlo
FlavorScheme: LHAPDF style
Flavors: [-5, -4, -3, -2, -1, 1, 2, 3, 4, 5]
NumFlavors: 5
XMin: 1e-05
XMax: 1
QMin: 1
QMax: 200
KtoQMin: 0.0001
```

```
KtoQMax: 2.001
```

while the info file produced for PV17 FF $D_1^{h\to\pi^+}$ is:

```
SetDesc: Set produced with NangaParbat + APFEL++ (please cite
   arXiv:1912.07550 and arXiv:1708.00911)
Authors: A. Bacchetta, F. Delcarro, C. Pisano, M. Radici, A.
   Signori
Reference: arXiv:1703.10157
SetIndex: 000000
SetName: PV17nll
TMDType: ff
CollDist: DSS14_NLO_Pip
CollDistMember: 0
Format: TMDlib2
DataVersion: 1
OrderQCD: NLL
AlphaS_OrderQCD: 0
Regularisation: bstarmin
NumMembers: 201
ErrorType: Monte Carlo
FlavorScheme: LHAPDF style
Flavors: [-5, -4, -3, -2, -1, 1, 2, 3, 4, 5]
NumFlavors: 5
XMin: 0.1
XMax: 1
QMin: 1
QMax: 100
KtoQMin: 0.0001
KtoQMax: 5
```

In `NangaParbat`, TMD grids have the format of a (`yaml`) map of vectors of vectors. This means that a grid is composed by objects with the structure:

$$\text{flavor}: \overbrace{\Bigg[\underbrace{\Big[[k_{t_1},\dots,k_{t_{\rm Nkt}}],\dots,[k_{t_1},\dots,k_{t_{\rm Nkt}}]\Big]}_{x_1},\dots,\underbrace{\Big[[k_{t_1},\dots,k_{t_{\rm Nkt}}],\dots,[k_{t_1},\dots,k_{t_{\rm Nkt}}]\Big]}_{x_{\rm Nx}}}^{Q_1}$$
$$,\ \dots\dots\ ,$$
$$\overbrace{\Bigg[\underbrace{\Big[[k_{t_1},\dots,k_{t_{\rm Nkt}}],\dots,[k_{t_1},\dots,k_{t_{\rm Nkt}}]\Big]}_{x_1},\dots,\underbrace{\Big[[k_{t_1},\dots,k_{t_{\rm Nkt}}],\dots,[k_{t_1},\dots,k_{t_{\rm Nkt}}]\Big]}_{x_{\rm Nx}}}^{Q_{\rm NQ}} \tag{3.74}$$

The key of the map is the flavor number, that, as shown in the information file, can go from -5 to 5 (we follow LHAPDF notation and the zero is skipped because it corresponds to the gluon).

A vector of vectors is associated to each flavor, *i.e.*, there is a structure like Eq. (3.74) for each flavor. In each vector of vectors the outer layer corresponds to different values of Q (in Eq. (3.74), $Q = Q_1, \ldots, Q_{\rm NQ}$), the second layer to values of x (or z, in the case of FFs, $x = x_1, \ldots, x_{\rm Nx}$) and the inner layer to values of k_T (or p_T for FFs, $k_t = k_{t_1}, \ldots, k_{t_{\rm Nkt}}$).

A detail concerning fragmentation functions is particularly worth of notice: usually, the FF D_1 is expressed with a dependence on $P_\perp$, the transverse momentum of final hadron within respect to virtual photon, as $D_1(z, P_\perp; Q)$, while in our grids is $D_1(z, p_\perp; Q)$, where $p_\perp$ is the transverse momentum of the parton. So, to obtain $D_1(z, P_\perp; Q)$ from our PV17 FFs grids, a change of variable is necessary:

$$P_\perp = -z p_\perp \tag{3.75}$$

To create the grids, a choice for $[Q_1, \ldots, Q_{\rm NQ}]$, $[x_1, \ldots, x_{\rm Nx}]$ and $[k_{t_1}, \ldots, k_{t_{\rm Nkt}}]$ has to be made: we used the points specified in `inc/NangaParbat/createtmdgrid.h` in the function `ThreeDGrid Inter3DGrid(std::string const& pf)`, which returns a *ThreeDGrid* object `grid` better tuned for PDFs or FFs according to the input string.

Grid points have been chosen through optimization tests that used PV17 and PV19 as functional forms: in order to determine the number of points and where such points should be in the Q, x, k_T spectrum, we performed several comparisons between the direct calculation of the TMD itself and the result of grid interpolation. More details about interpolation tools and their accuracy are given in Sec. 3.9, but a few more words on the motivation of our choices are appropriate here.

The number of points to put in the grids has been chosen taking into account also the size of the files that are produced, as this is a very important quality for a product that to be used has to be easy to share. We managed to obtain a good compromise between the size of the files and the accuracy of the grids, and a TMD PDF grid (for a single replica) is $\sim$ 16 MB, while a FF grid is $\sim$ 10 MB (for each replica).

As for the choice of the values of Q, x and k_T to grid, the selection of the specific points may be dependent on the functional form of the extraction that is tabulated. We found that the values reported above work well for both PV17 and PV19, which present different functional forms; however, should `NangaParbat` be used for the production of grids for different extractions, we advise the users to run a few test on the accuracy of the interpolation to see if the grid points in `inc/NangaParbat/createtmdgrid.h` suite their needs.

Note that in the grids the k_T points are expressed as k_T/Q: this is more convenient

within respect the use of a simple vector in k_T, as it allows us to specify only once the spacing of the points over the range in k_T, while if we put specific values for k_T we would need to have a vector of values of k_T for every value of Q.

If the necessity to have different kinematic points in TMD grids arises, the required changes to the standard choice (which is the one present above) can be done directly in `inc/NangaParbat/createtmdgrid.h`, or, a better option that does not disrupt the `NangaParbat` structure is to create another function, equivalent to `ThreeDGrid Inter3DGrid(std::string const& pf)`, which contains the desidered Q, x, k_T values, and then use such function in `NangaParbat::ProduceTMDGrid` in `src/tmdgrid/createtmdgrid.cc`.

The values of the kinematic points in the grids do not matter for the interpolation: the crucial characteristic that all grids must have to be used with the `NangaParbat` interpolator is the `YAML` structure, which must be the one shown in Eq. (3.74). The kinematic points we chose for TMD grids are reported in Appendix 3.D.

As mentioned above, the total size of the bundle of grid files is particularly important for TMD delivery: keeping a reasonable size of the grids is mandatory for the release of a TMD ensemble, which is typically made of one grid for each TMD replica available. We tackled the problem of storage space by selecting carefully the interpolating grid, in such a way that a good accuracy is preserved without increasing too much the number of interpolating points. Other solutions to this crucial point have been conceived: for example, in Ref. [195], the authors study a compression algorithm for Monte Carlo PDF sets. The scope of such algorithm is to find a subset of the original set of replicas such that delivers a probability distribution as indistinguishable from the one of the whole set of replicas set as possible. In particular, Ref. [195] shows that, with the use of an optimal compression strategy, it is possible to obtain an ensemble of $N_{\rm rep}$ replicas with the same statistical features of the $\tilde{N}_{\rm rep} \gg N_{\rm rep}$ original replicas with sufficient accuracy for most relevant application.

3.8.2 Structure function grids

To compute the SIDIS cross section, the structure function $F_{\rm UU,T}$ is needed. The structure function $F_{\rm UU,T}$ is a convolution a TMD PDF, f_1, and a TMD FF, D_1. Ideally, a user could compute the structure function directly, by interpolating the two appropriate TMD grids and convolute them using the tools provided by `NangaParbat`. However, for experimental studies it could be preferable to have directly $F_{\rm UU,T}$ grids. Therefore, we made them available in `NangaParbat`.

We produced $F_{\rm UU,T}$ grids starting from TMDs extracted in PV17 [84], since in PV19 we analyzed only Drell–Yan data and thus we did not have access to the TMD FFs D_1. For the moment, we provide structure functions for the production of π^+ and

π^-, denoted as $F_{\rm UU,T}^{h\to\pi^+}$ and $F_{\rm UU,T}^{h\to\pi^-}$. In the future, other final–state hadrons will be included.

The unpolarized structure function could be directly computed in b_T–space, as done during the fit. However, using TMD grids has the advantage of avoiding some problems that could arise if TMD PDFs and TMD FFs are treated in different ways.

The `NangaParbat` implementation of structure functions grid production is very similar to that of TMDs: `run/CreateStructGrids` calls the function `NangaParbat::ProduceStructGrid`, implemented in `src/tmdgrid/createstructgrid.cc`, that computes a four dimensional grid (in x, z, q_T and Q).

To run, `run/CreateStructGrids` needs the following inputs:

```
./CreateStructGrids <main fit directory with TMD grids> <name
   of TMD PDF set> <name of TMD FF set> <output> <[optional]
   replica ID>
```

In particular, the names of the two `NangaParbat` TMD grids are needed, one for the PDF and one for the FF. This choice has been made because an incompatibility of quark thresholds in the two different collinear sets used (MMHT2014lo68cl for PDFs and DSS14NLO for FFs) caused problems in the definition of the active flavors at certain values of Q (specifically between $Q = 4.3$GeV, mass of the bottom in DSS14NLO, and $Q = 4.75$GeV mass of the bottom in MMHT2014lo68cl). Due to how the direct calculation of $F_{\rm UU,T}$ is implemented in `NangaParbat`, these problems arise when performing the computation in b_T–space, while they are not present when convoluting two TMD grids.

The tool that performs the convolution, implemented in `src/tmdgrid/createtmdgrid.cc` is discussed in Sec. 3.9.3.

Analogously to TMD grids, also in the structure function case an information file is produced by the function `NangaParbat::EmitStructInfo`:

```
SetDesc: Set produced with NangaParbat + APFEL++ (please cite
   arXiv:1912.07550 and arXiv:1708.00911)
Authors: A. Bacchetta, F. Delcarro, C. Pisano, M. Radici, A.
   Signori
Reference: arXiv:1703.10157
SetIndex: 000000
StructFuncType: FUUT
Target: proton
FinalStateHadron: Pip
TMDPDF: PV17nll
TMDFF: PV17nll
Format: TMDlib2
DataVersion: 1
OrderQCD: NLL
NumMembers: 200
```

```
ErrorType: Monte Carlo
NumFlavors: 5
XMin: 1e-05
XMax: 0.8
ZMin: 0.1
ZMax: 0.8
QMin: 1
QMax: 100
KtoQMin: 0.0001
KtoQMax: 1
```

The C++ structure that supports the production of four–dimensional grids in `NangaParbat` is `struct FourDGrid`, in `inc/NangaParbat/createstructgrid.h`.

The structure of $F_{\mathrm{UU,T}}$ grids can be deduced from the function that creates them, `NangaParbat::EmitStructGrid` in `src/tmdgrid/createstructgrid.cc` where the object

```
std::vector<std::vector<std::vector<std::vector<double>>>> SFs
```

is filled following the indexes order `SFs[iQ][ix][iz][iqT]`:

$$\mathrm{SFs}[Q_1, \ldots, Q_{\mathrm{NQ}}][x_1, \ldots, x_{\mathrm{Nx}}][z_1, \ldots, z_{\mathrm{Nz}}][q_{t_1}, \ldots, q_{t_{\mathrm{Nkt}}}] \tag{3.76}$$

Since to obtain $F_{\mathrm{UU,T}}$ there is a sum over all the flavors, the structure of these grids is simply in the format of vectors of vectors, without the flavor map (needed for TMDs). With this exception, the structure of $F_{\mathrm{UU,T}}$ is analogous to the one in Eq. (3.74) with the addition of the index z, as shown in Eq. (3.76).

Also in the case of $F_{\mathrm{UU,T}}$ grids, we performed tests optimized the choice kinematic points in view of the use of such grids with the `NangaParbat` interpolator. For more details about these tests see also the following section, Sec. 3.9. The kinematic points we included in the grids are reported in Appendix 3.D. Currently $F_{\mathrm{UU,T}}$ grids have a size of 31 MB and there are 200 replicas available.

3.9 Tools

In order to best exploit the information contained in our grids, we created a few tools to use them. These tools are contained in the `tools/` section of `NangaParbat` and consist in functions that interpolate three–dimensional and four–dimensional grids and convolute two TMD grids.

The interpolation of the grids can be done with `TMDGridInterpolation.cc` for TMDs and with `tools/StructGridInterpolation.cc` for structure functions, while `GridsConvolution.cc` performs the convolution between a TMD PDF grid and a

TMD FF grid. The input files, where the user can choose the kinematic points for the interpolation and the convolution, are in `tools/inputs/`.

`NangaParbat` makes use of the `APFEL++` interpolator to get results from grids: in Sec. 3.9.1, a brief overview of the formulas implemented in `APFEL++` is given, as well as some plots of the tests we run on the grid interpolation.

In Sec. 3.9.3 the convolution tool is discussed.

3.9.1 Interpolation

Interpolation is an extremely powerful tool that allows to reconstruct, within some accuracy, continuous functions when knowing them only in a finite number of points. Unsurprisingly, interpolation techniques are massively used in numerical codes.

In order to make the best of interpolation, `APFEL++` has been designed around a specific choice of the interpolation strategy. Specifically, the very computational core of `APFEL++` relies on *Langrange* polynomials and their properties. Despite Langrange polynomials do not enjoy the smoothness of splines, they enjoy a number of very useful properties that allow extending the use of Lagrange polynomials from just interpolation to derivation and integration.

Lagrange interpolation

In this section we will derive a general expression for the Lagrange interpolating functions w.

Suppose one wants to interpolate the test function g in the point x using a set of Lagrange polynomials of degree k. This requires a subset of $k+1$ consecutive points on an interpolation grid, for example $\{x_\alpha, \ldots, x_{\alpha+k}\}$. The relative position between the point x and the subset of points used for the interpolation is arbitrary. It is convenient to choose the subset of points such that $x_\alpha < x \leq x_{\alpha+k}$. [27] However, the ambiguity remains because there are k possible choices according to whether $x_\alpha < x \leq x_{\alpha+1}$, or $x_{\alpha+1} < x \leq x_{\alpha+2}$, and so on. For now we assume that:

$$x_\alpha < x \leq x_{\alpha+1}\,, \tag{3.77}$$

but this assumption can be released later. Using the standard Lagrange interpolation

[27] Actually, it is not necessary to impose the constraint $x_\alpha < x \leq x_{\alpha+k}$. In case this relation is not fulfilled one usually speaks about *extrapolation* rather than *interpolation*. If not necessary, this option is typically not convenient because it may lead to a substantial deterioration in the accuracy with which $g(x)$ is determined.

procedure, one can approximate the function g in x as:

$$g(x) = \sum_{i=0}^{k} \ell_i^{(k)}(x) g(x_{\alpha+i}) \,, \tag{3.78}$$

where $\ell_i^{(k)}$ is the i–th Lagrange polynomial of degree k which can be written as:

$$\ell_i^{(k)}(x) = \prod_{\substack{m=0 \\ m\neq i}}^{k} \frac{x - x_{\alpha+m}}{x_{\alpha+i} - x_{\alpha+m}} \,. \tag{3.79}$$

Since we have assumed that $x_\alpha < x \leq x_{\alpha+1}$ (see Eq. (3.77)), Eq. (3.78) can be written as:

$$g(x) = \theta(x - x_\alpha)\theta(x_{\alpha+1} - x) \sum_{i=0}^{k} g(x_{\alpha+i}) \prod_{\substack{m=0 \\ m\neq i}}^{k} \frac{x - x_{\alpha+m}}{x_{\alpha+i} - x_{\alpha+m}} \,. \tag{3.80}$$

In order to make Eq. (3.80) valid for all values of α, one just has to sum over all $N_x + 1$ nodes of the *global* interpolation grid $\{x_0, \ldots, x_{N_x}\}$, that is:

$$g(x) = \sum_{\alpha=0}^{N_x-1} \theta(x - x_\alpha)\theta(x_{\alpha+1} - x) \sum_{i=0}^{k} g(x_{\alpha+i}) \prod_{\substack{m=0 \\ m\neq i}}^{k} \frac{x - x_{\alpha+m}}{x_{\alpha+i} - x_{\alpha+m}} \,. \tag{3.81}$$

Defining $\beta = \alpha + i$, one can rearrange the equation above as:

$$g(x) = \sum_{\beta=0}^{N_x+k-1} w_\beta^{(k)}(x) g(x_\beta) \,, \tag{3.82}$$

that leads to the definition of the interpolating functions:

$$w_\beta^{(k)}(x) = \sum_{\substack{i=0 \\ i\leq\beta}}^{k} \theta(x - x_{\beta-i})\theta(x_{\beta-i+1} - x) \prod_{\substack{m=0 \\ m\neq i}}^{k} \frac{x - x_{\beta-i+m}}{x_\beta - x_{\beta-i+m}} \,. \tag{3.83}$$

Notice that the condition $i \leq \beta$ comes from the condition $\alpha \geq 0$. It is important to observe that the sum in Eq. (3.82) extends up to the $(N_x + k - 1)$–th node. Therefore, the original grid needs to be extended by $k - 1$ nodes. However, the range of validity of the interpolation remains that defined by the original grid, *i.e.* $x_0 \leq x \leq x_{N_x}$.

Typically, only a small number of terms in the sum in Eq. (3.82) is different from zero. For any given value of x, it is possible to determine the values of the index β for which the interpolating functions $w_\beta^{(k)}$ are different from zero, reducing (often

dramatically) the amount of sums required to carry out an interpolation. The range of β is easily determined by observing that in Eq. (3.80) the summation extends on the nodes between x_α and $x_{\alpha+k}$. But since β is defined like $\alpha + i$ this exactly defines the range in β:

$$\alpha(x) \leq \beta \leq \alpha(x) + k\,, \tag{3.84}$$

where the function $\alpha(x)$ is implicitly defined through Eq. (3.77). Therefore, Eq. (3.82) becomes:

$$g(x) = \sum_{\beta=\alpha(x)}^{\alpha(x)+k} w_\beta^{(k)}(x) g(x_\beta)\,, \tag{3.85}$$

Since interpolation functions $w_\beta^{(k)}(x)$ often appear inside integrals, it is very useful to use the fact that they are different from zero only over a limited interval, specifically:

$$w_\beta^{(k)}(x) \neq 0 \quad \Leftrightarrow \quad x_{\beta-k} < x < x_{\beta+1}\,. \tag{3.86}$$

This allows one to optimize integrations, such as the one to obtain the weights in Eq. (3.42), restricting the integration region only to where the interpolating functions are different from zero.

Generalized interpolation

Sometimes it may be useful to release this assumption: for example, a situation in which this is advantageous is in the presence of non-smooth or discontinuous functions (such as PDFs and FFs as function of the factorization scale μ in correspondence of the heavy-quark thresholds). When interpolating these functions one should not interpolate over the discontinuities. To do so and yet retain a given interpolation degree, one can release the assumption in Eq. (3.77). Specifically, we generalize it to:

$$x_{\alpha+t} < x \leq x_{\alpha+t+1} \quad \text{with} \quad t = 0, \ldots, k-1\,, \tag{3.87}$$

such that the interpolation formula becomes:

$$g(x) = \sum_{\alpha=-t}^{N_x-t-1} \theta(x - x_{\alpha+t})\theta(x_{\alpha+t+1} - x) \sum_{i=0}^{k} g(x_{\alpha+i}) \prod_{\substack{m=0 \\ m\neq i}}^{k} \frac{x - x_{\alpha+m}}{x_{\alpha+i} - x_{\alpha+m}}\,, \tag{3.88}$$

that can be rearranged as:

$$g(x) = \sum_{\beta=-t}^{N_x+k-t-1} w_{\beta,t}^{(k)}(x) g(x_\beta)\,, \tag{3.89}$$

with:

$$w_{\beta,t}^{(k)}(x) = \sum_{i=0,i\leq\beta}^{k} \theta(x - x_{\beta-i+t})\theta(x_{\beta-i+t+1} - x) \prod_{m=0,m\neq i}^{k} \frac{x - x_{\beta-i+m}}{x_\beta - x_{\beta-i+m}}\,, \tag{3.90}$$

being the generalized interpolation functions. We observe that the support region of $w_{\beta,t}^{(k)}$ is:

$$w_{\beta,t}^{(k)}(x) \neq 0 \quad \Leftrightarrow \quad x_{\beta+t-k} < x < x_{\beta+t+1}\,, \tag{3.91}$$

that generalizes Eq. (3.86). The generalized interpolation functions can be used to avoid interpolating over some particular grid nodes. In order to avoid interpolation over a specific node of the grid, t can be chosen dynamically in such a way that $\beta + t$ in Eq. (3.90) never corresponds to that particular node.

This mechanism is implemented (in `APFEL++`) as follows. The interpolation grid is chosen to have two nodes in correspondence of the *threshold* x_T, but slightly displaced up and down by an "infinitesimal" amount ϵ to keep them separate, that is:

$$\{x_0, \ldots, x_{\alpha_t-1}, x_{\alpha_t}, \ldots, x_{N_x}\} \quad \text{with} \quad x_{\alpha_t-1} = x_T - \epsilon \quad \text{and} \quad x_{\alpha_t} = x_T + \epsilon\,. \tag{3.92}$$

The aim is then to avoid interpolating over the nodes x_{α_t-1} and x_{α_t}. By default we assume $t = 0$ in Eq. (3.90) so that we automatically reduce to Eq. (3.83). In this situation, we are assuming Eq. (3.77) where effectively the index α is determined dynamically according to the values of x. Therefore, we can effectively write:

$$x_{\alpha(x)} < x \leq x_{\alpha(x)+1}\,, \tag{3.93}$$

which implicitly defines the function $\alpha(x)$. Eq. (3.80) then requires summing over the $k+1$ nodes of the grid $\{x_{\alpha(x)}, \ldots, x_{\alpha(x)+k}\}$. However, when the point x approaches x_T from below, the range $\{x_{\alpha(x)}, \ldots, x_{\alpha(x)+k}\}$ may end up enclosing both nodes x_{α_t-1} and x_{α_t}. To avoid this, we promote the index t in Eq. (3.87) to a function of x defined through the inequalities:

$$\begin{cases} x < x_T\,, \\ \\ x_{\alpha_t-2} < x_{\alpha(x)-t(x)+k} \leq x_{\alpha_t-1}\,, \end{cases} \tag{3.94}$$

that translate into:

$$\begin{cases} \alpha(x) \leq \alpha_t - 2\,, \\ \\ \alpha_t - 2 < \alpha(x) - t(x) + k \leq \alpha_t - 1\,. \end{cases} \tag{3.95}$$

Imposing the unnecessary but convenient constraint $t(x) \geq 0$, finally gives:

$$t(x) = \max\left[\min\left[\alpha(x), \alpha_t - 2\right] - \alpha_t + k + 1, 0\right] , \tag{3.96}$$

that also obeys:

$$0 \leq t(x) \leq k - 1 \, , \tag{3.97}$$

as required. In addition, as in Eq. (3.85), the summation over β in Eq. (3.89) can be restricted to a range of $k + 1$ nodes as:

$$g(x) = \sum_{\beta=\alpha(x)-t(x)}^{\alpha(x)-t(x)+k} w_{\beta,t}^{(k)}(x) g(x_\beta) \, . \tag{3.98}$$

3.9.2 Grid interpolation tests

We tested the grid we produced with `NangaParbat` (Sec. 3.8) with the in interpolator based on `APFEL++` described above. To test TMD grids (and the interpolator) we used the program `tests/GridProduction.cc`, which compares results obtained from the grids with the direct computation of the TMD in b_T–space. The four–dimensional interpolator and the $F_{\rm UU,T}$ grids have been tested with `tests/FUUTGridProduction.cc`, analogous to the TMD one.

The results of the tests are shown in plots in Appendix 3.E. We iteratively modified the number of points to include in the grids and the spacing between them until the relative discrepancy between direct computation and interpolation from the grids was below 2%.

The program `tools/TMDGridInterpolation` allows the user to interpolate `NangaParbat` TMD grids, which have to be downloaded separately (they are not included in the `NangaParbat` framework, but they can be downloaded at the link reported in the introduction of this section). The program is executed with the command:

```
./TMDGridInterpolation <grid main folder> <grid name> <n. repl
   .> <output>
```

where the first input is the relative path to the folder where the user has stored the grids, the second input consists in the name of the grid set to interpolate. Then, the number of the replicas that the user wants to consider is specified and also a name for the output folder, where the values of the TMD in the desired points will be stored, is to be indicated.

In `tools/TMDGridInterpolation.cc`, grids are initialized with

```
NangaParbat::TMDGrid* TMDs = NangaParbat::mkTMD(Name, Folder,
   ReplicaNumber);
```

and interpolated with `TMDs->Evaluate(x , kToQ * Q , Q).at(ifl)`, where `ifl` is a number that indicates the flavor that the user wants to consider and x, $kToQ$ and Q are read form the input card in `tools/inputs/TMDGridInterpolation.yaml` which the users can modify to suit their needs. The program `tools/StructGridInterpolation.cc` works in a similar way, and the function `Evaluate` in this case calls the four–dimensional interpolator.

The output of both programs is a `yaml` file.

As a final remark it is important to stress that the interpolator called in `tools/TMDGridInterpolation.cc` and `tools/StructGridInterpolation.cc` does not necessarily need a grid created with `NangaParbat`: grids created with any program that have the same structure of the grids illustrated in Sec. 3.8 are compatible with the `NangaParbat` interpolator.

3.9.3 Convolution

Besides the interpolator, `NangaParbat` offers another tool to manipulate grids: in order to compute structure functions, a convolution between two TMDs is required, and this operation is done by the program `tools/GridsConvolution.cc`, which is run in the following way:

```
./GridsConvolution <directory with TMD grids> <name of TMD
   PDF set> <name of TMD FF set> <replica ID>
```

where, as in `tools/TMDGridInterpolation.cc`, `<directory with TMD grids>` is the relative path to the folder where the grids are, the second and third inputs are the names of the grids to convolute and `<replica ID>` is the replica to consider.

The convolution between two TMD grids is implemented in `src/tmdgrid/factories.cc` in the function:

```
/**
 * @brief Function that performs the convolution of two TMD
 * distributions in kT space.
 * @param TMD1: first distribution
 * @param TMD2: second distribution
 * @param Charges: to be used as weights of the partonic
    combinations
 * @param kTCutOff: cutoff on the integration in kT relative
    to Q (default: 1)
 * @param IntEps: integration relative accuracy (default: 1e
    -5)
 */
std::function<double(double const&, double const&, double
   const&, double const&)>
Convolution(TMDGrid          const* TMD1,
```

```
12              TMDGrid           const* TMD2,
13              std::function<std::vector<double>(double const&)>
                   const& Charges,
14              double            const& kTCutOff = 1,
15              double            const& IntEps = 1e-5);
```

The formula encoded in `NangaParbat::Convolution` is:

$$\mathrm{TMD}_1 \otimes \mathrm{TMD}_2 = \sum_q c_q(Q) \times \int_0^\infty dk_T\, k_T \int_0^{2\pi} d\theta\, \mathrm{TMD}_1(x_1, k_T; \mu, \zeta)\mathrm{TMD}_2(x_2, \sqrt{q_T^2 + k_T^2 - 2q_T k_T \cos\theta}; \mu, \zeta) \tag{3.99}$$

and takes into account if the convolution is computed between two distributions of the same type (as for Drell–Yan and e^+e^- annihilation) or between one TMD PDF and one TMD FF (SIDIS case).

```
1  // Changes q and qbar in order to properly compute the
       luminosity.
2  int sgn = 1; // SIDIS
3  if (TMD1->GetInfoNode()["TMDType"].as<std::string>() == TMD2->
       GetInfoNode()["TMDType"].as<std::string>())
4    sgn = -1;  // DY, e+e-
5
6  return [=] (double const& x1, double const& x2, double const&
       Q, double const& qT) -> double
7  {
8    const std::vector<double> Bq = Charges(Q);
9    apfel::Integrator integrandKT{
10     [=] (double const& kT) -> double
11     {
12       apfel::Integrator integrandTheta{
13         [=] (double const& theta) -> double
14         {
15           const std::map<int, double> d1 = TMD1->Evaluate(x1,
                 kT, Q);
16           const std::map<int, double> d2 = TMD2->Evaluate(x2,
                 sqrt( pow(kT, 2) + pow(qT, 2) - 2 * kT * qT * cos
                 (theta) ), Q);
17           double lumi = 0;
18           for (int i = 1; i <= 5; i++)
19             lumi += Bq[i-1] * ( d1.at(i) * d2.at(sgn * i) + d1
                   .at(-i) * d2.at(-sgn * i) );
20           return lumi;
21         }
22       };
```

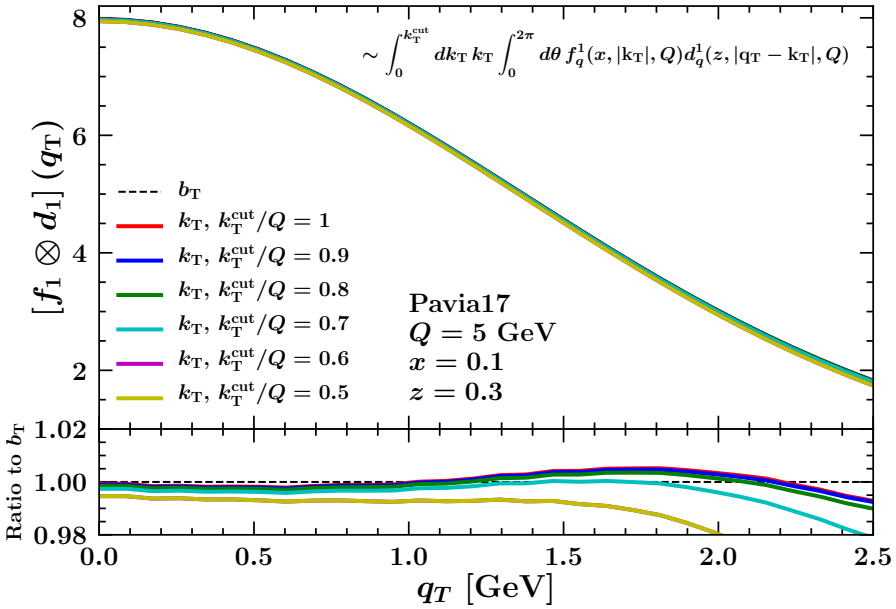


Figure 3.2: Dependence of the numerically computed convolution formula on the k_T space cut–off.

```
23        return kT * integrandTheta.integrate(0, 2 * M_PI, IntEps);
24        }
25    };
26    return 2 * M_PI * integrandKT.integrate(0, kTCutOff * Q,
          IntEps);
```

In particular, when we convolute f_1 and D_1, we have:

$$F_{\mathrm{UU,T}}(x, z, q_T, Q) = \sum_q c(Q) f_1^q \otimes D_1^{q\to\pi^\pm}(x, z, q_T, Q) \tag{3.100}$$

Our $F_{\mathrm{UU,T}}^{h\to\pi^\pm}$ grids have been obtained performing the convolution between our PV17 f_1 and $D_1^{h\to\pi^\pm}$ grids.

Note that in Eq. (3.99) the integral in k_T extends up to infinity. In practice, such integral in order to be computed numerically needs a cut–off over the momenta. For its numerical computation, a k_T cut–off is applied. It follows that the implementation of Eq. (3.99) has a dependence on such cut–off, as it is shown in Fig. 3.2 where we can observe that, at $Q = 5$ GeV, truncating the integral too soon, at $k_T^{\mathrm{cut}}/Q = 0.5$, for example, can result in a discrepancy of more than 2% between the integral computed numerically with Eq. (3.99) and the same integral in computed in b_T–space.

Therefore, discrepancies between direct calculation and numerical integration can be significant, even though, looking at the superimposed curves in the upper panel of Fig. 3.2, it might seem otherwise. In particular, considering that even with $k_T^{\mathrm{cut}}/Q = 1$

there is a loss of $\sim 1\%$ and that the discrepancies vary with the kinematic region in which the convolution is computed, to produce our $F_{\rm UU,T}$ grids we adopted the conservative choice $k_T^{\rm cut}/Q = 5$.

3.A Appendix - Cuts on the final-state leptons

In this section, we derive the explicit expression of the phase-space reduction factor $\mathcal{P}$. This factor is defined as[28]

$$\mathcal{P}(q) = \frac{\displaystyle\int_{\text{fid. reg.}} d^4p_1 d^4p_2\, \delta(p_1^2)\delta(p_2^2)\theta(p_{1,0})\theta(p_{2,0})\delta^{(4)}(p_1+p_2-q)L_\perp(p_1,p_2)}{\displaystyle\int d^4p_1 d^4p_2\, \delta(p_1^2)\delta(p_2^2)\theta(p_{1,0})\theta(p_{2,0})\delta^{(4)}(p_1+p_2-q)L_\perp(p_1,p_2)}\,, \tag{3.101}$$

where p_1 and p_2 are the four-momenta of the outgoing leptons. The integral in the numerator extends over the fiducial region defined by the cuts on the final-state leptons. The quantity $L_\perp$ is defined as

$$L_\perp = g_\perp^{\mu\nu} L_{\mu\nu}\,, \tag{3.102}$$

where $L_{\mu\nu}$ is the (parity-conserving part of the) *leptonic tensor* that, assuming massless leptons, reads

$$L^{\mu\nu} = 4(p_1^\mu p_2^\nu + p_2^\mu p_1^\nu - g^{\mu\nu} p_1 p_2)\,, \tag{3.103}$$

while the transverse metric is given by

$$g_\perp^{\mu\nu} = g^{\mu\nu} + z^\mu z^\nu - t^\mu t^\nu\,. \tag{3.104}$$

The vectors z^μ and t^μ, in the Collins-Soper frame, are defined as

$$z^\mu = (\sinh y, \mathbf{0}, \cosh y)\,, \qquad t^\mu = \frac{q^\mu}{Q}\,, \tag{3.105}$$

and they are such that $z^2 = -1$, $t^2 = 1$ and $(z\cdot q) = 0$. The effect of integrating over the fiducial region in the numerator of Eq. (3.101) can be implemented by defining a generalised θ-function, $\Phi(p_1, p_2)$, that is equal to one inside the fiducial region and zero outside. This allows one to integrate also the numerator over the full phase-space of the two outgoing leptons. Next, we integrate out one of the momenta, say p_2, exploiting

[28] In Eq. (3.101) a parity-violating term is neglected. We will argue in Sec. 3.A.1 that its contribution is negligible for realistic cuts.

the momentum-conservation δ-function:

$$P(q) = \frac{\int d^4p\delta(p^2)\delta((q-p)^2)\theta(p_0)\theta(q_0-p_0)L_\perp(p,q-p)\Phi(p,q-p)}{\int d^4p\delta(p^2)\delta((q-p)^2)\theta(p_0)\theta(q_0-p_0)L_\perp(p,q-p)}\,, \tag{3.106}$$

where we have renamed $p = p_1$. The remaining δ-functions can be used to constrain two of the four components of the momentum p. The first, $\delta(p^2)$, is typically used to set the energy component of p, p_0, on the mass shell. Since the leptons are massless, this produces

$$\int d^4p\delta(p^2)\theta(p_0) = \int d^4p\,\delta(p_0^2-|\mathbf{p}|^2)\theta(p_0) = \int \frac{dp_0 d^3\mathbf{p}}{2|\mathbf{p}|}\delta(p_0-|\mathbf{p}|) = \int \frac{d^3\mathbf{p}}{2|\mathbf{p}|}\,. \tag{3.107}$$

Of course, the four-momentum p appearing in the rest of the integrand has to be set on shell ($p_0 = |\mathbf{p}|$). Now we express the three-dimensional measure $d^3\mathbf{p}$ in terms of the transverse momentum $\mathbf{p}_T$, the pseudo-rapidity η, and the azimuthal angle ϕ of the lepton:

$$\int \frac{d^3\mathbf{p}}{2|\mathbf{p}|} = \int \frac{d|\mathbf{p}_T|^2}{4} d\eta\, d\phi\,. \tag{3.108}$$

Now we consider the second δ-function, $\delta((q-p)^2)$, in Eq. (3.106). It is convenient to express the vectors q and p in terms of the respective invariant mass, pseudo-rapidity, and transverse momentum:

$$\begin{aligned} q &= (M\cosh y, \mathbf{q}_T, M\sinh y)\ , \\ p &= (|\mathbf{p}_T|\cosh\eta, \mathbf{p}_T, |\mathbf{p}_T|\sinh\eta)\ , \end{aligned} \tag{3.109}$$

with $M = \sqrt{Q^2+|\mathbf{q}_T|^2}$. Without loss of generality, we assume that the two-dimensional vector $\mathbf{q}_T$ is aligned with the x axis so that $\mathbf{p}_T\cdot\mathbf{q}_T = |\mathbf{p}_T||\mathbf{q}_T|\cos\phi$.[29] This leads to

$$\delta((q-p)^2) = \delta\left(Q^2 - 2|\mathbf{p}_T|\left[M\cosh(\eta-y) - |\mathbf{q}_T|\cos\phi\right]\right)\,, \tag{3.110}$$

[29] In the general case in which $\mathbf{q}_T$ forms an angle β with the x axis, the scalar product would result in $|\mathbf{p}_T||\mathbf{q}_T|\cos(\phi-\beta)$. However, for observables inclusive in azimuthal angle, the angle β can always be reabsorbed in a redefinition of ϕ.

so that

$$\mathcal{P}(q) = \frac{\displaystyle\int \frac{d|\mathbf{p}_T|^2}{4} d\eta\, d\phi\, \delta\left(Q^2 - 2|\mathbf{p}_T|\left[M\cosh\left(\eta - y\right) - |\mathbf{q}_T|\cos\phi\right]\right) L_\perp(p, q-p)\Phi(p, q-p)}{\displaystyle\int \frac{d|\mathbf{p}_T|^2}{4} d\eta\, d\phi\, \delta\left(Q^2 - 2|\mathbf{p}_T|\left[M\cosh\left(\eta - y\right) - |\mathbf{q}_T|\cos\phi\right]\right) L_\perp(p, q-p)}\,, \tag{3.111}$$

where the vector p is understood to be on-shell. Now we compute $L_\perp(p, q-p)$ contracting $L_{\mu\nu}$ in Eq. (3.103) with the transverse metric $g_\perp^{\mu\nu}$ in Eq. (3.104) using Eq. (3.109):

$$L_\perp(p, q-p) = 2Q^2\left[1 + 4\sinh^2(y-\eta)\frac{|\mathbf{p}_T|^2}{Q^2}\right]. \tag{3.112}$$

We can now integrate out one of the variables in the integrals in Eq. (3.111) by making use of the remaining δ-function. Somewhat counterintuitively, it is convenient to integrate over $|\mathbf{p}_T|$. This produces

$$P(q) = \frac{\displaystyle\int_{-\infty}^{\infty} d\eta \int_0^{2\pi} d\phi \left[\frac{2\overline{p}_T^2}{Q^2} + 2\sinh^2(y-\eta)\frac{\overline{p}_T^4}{Q^4}\right]\Phi(\overline{p}, q-\overline{p})}{\displaystyle\int_{-\infty}^{\infty} d\eta \int_0^{2\pi} d\phi \left[\frac{2\overline{p}_T^2}{Q^2} + 2\sinh^2(y-\eta)\frac{\overline{p}_T^4}{Q^4}\right]}\,, \tag{3.113}$$

where $\overline{p}_T$ is defined as

$$\overline{p}_T = \frac{Q^2}{2|\mathbf{q}_T|}\frac{1}{\left[\frac{M\cosh(\eta-y)}{|\mathbf{q}_T|} - \cos\phi\right]}\,. \tag{3.114}$$

and $\overline{p}$ symbolises the on-shell vector p with the absolute value of the transverse component set equal to Eq. (3.114). Next we turn to consider the integral in ϕ. To this end, the following relation

$$\int_0^{2\pi} d\phi\, f(\cos\phi) = \int_{-1}^{1}\frac{dx}{\sqrt{1-x^2}}\left[f(x) + f(-x)\right], \tag{3.115}$$

along with the *indefinite* integrals

$$\int\frac{dx}{(a\pm x)^2\sqrt{1-x^2}} = \frac{\sqrt{1-x^2}}{(a^2-1)(x\pm a)} \pm \frac{a}{(a^2-1)^{3/2}}\tan^{-1}\left(\frac{1\pm ax}{\sqrt{a^2-1}\sqrt{1-x^2}}\right), \tag{3.116}$$

and

$$\int \frac{dx}{(a\pm x)^4\sqrt{1-x^2}} = \frac{\sqrt{1-x^2}\,[(11a^2+4)x^2 \pm 3a(9a^2+1)x + (18a^4-5a^2+2)]}{6(a^2-1)^3(x\pm a)^3} \pm \frac{a(2a^2+3)}{2(a^2-1)^{7/2}} \tan^{-1}\left(\frac{1\pm ax}{\sqrt{a^2-1}\sqrt{1-x^2}}\right), \tag{3.117}$$

enable us to compute analytically the primitive function of the integrals in ϕ in Eq. (3.113). Eqs. (3.116) and (3.117) are particularly useful because they allow us to compute the integral over ϕ analytically also in the presence of cuts. Let us first compute the integral in the denominator of Eq. (3.113), *i.e.* the integral of $L_\perp$ over the full phase-space. To do so, using Eqs. (3.116) and (3.117), we compute the following definite integrals

$$\int_{-1}^{1} \frac{dx}{(a\pm x)^2\sqrt{1-x^2}} = \frac{\pi a}{(a^2-1)^{3/2}}, \tag{3.118}$$

and:

$$\int_{-1}^{1} \frac{dx}{(a\pm x)^4\sqrt{1-x^2}} = \frac{\pi a(2a^2+3)}{2(a^2-1)^{7/2}}. \tag{3.119}$$

Using these results, and finally integrating over η, gives the well-known result

$$\int d^4p_1 d^4p_2\, \delta(p_1^2)\delta(p_2^2)\theta(p_{1,0})\theta(p_{2,0})\delta^{(4)}(p_1+p_2-q)L_\perp(p_1,p_2) = \frac{4\pi}{3}Q^2. \tag{3.120}$$

In order to compute the numerator of Eq. (3.113), we need to insert the appropriate function Φ. Typically, in DY production the kinematic cuts are imposed independently on the same variables for both the final-state leptons. Therefore, the function Φ factorises into two identical functions acting on each lepton momentum:

$$\Phi(p_1,p_2) = \Theta(p_1)\Theta(p_2). \tag{3.121}$$

We are specifically interested in kinematic cuts on the rapidity and on the transverse momentum of the following kind

$$\eta_{\min} < \eta_{1(2)} < \eta_{\max} \quad \text{and} \quad |\mathbf{p}_{T,1(2)}| > p_{T,\min}. \tag{3.122}$$

Therefore

$$\Theta(p) = \vartheta(\eta-\eta_{\min})\vartheta(\eta_{\max}-\eta)\vartheta(|\mathbf{p}_T|-p_{T,\min}). \tag{3.123}$$

Using Eqs. (3.109) and (3.114) gives

$$\begin{aligned}\Phi(p,q-p) &= \vartheta(\eta-\eta_{\min})\times\vartheta(\eta_{\max}-\eta)\\ &\times\ \vartheta(\cos\phi - f^{(2)}(\eta,p_{T,\min}))\\ &\times\ \vartheta(f^{(3)}(\eta,\eta_{\min})-\cos\phi)\times\vartheta(f^{(3)}(\eta,\eta_{\max})-\cos\phi)\\ &\times\ \vartheta(f^{(4)}(\eta,p_{T,\min})-\cos\phi)\,,\end{aligned} \tag{3.124}$$

with

$$\begin{aligned}f^{(2)}(\eta,p_{T,\text{cut}}) &= \frac{2Mp_{T,\text{cut}}\cosh(\eta-y)-Q^2}{2p_{T,\text{cut}}|\mathbf{q}_T|}\,,\\ f^{(3)}(\eta,\eta_{\text{cut}}) &= \frac{M\cosh(\eta-y)}{|\mathbf{q}_T|}-\frac{Q^2\left(\sinh(\eta-y)\coth(y-\eta_{\text{cut}})+\cosh(\eta-y)\right)}{2|\mathbf{q}_T|M}\,,\\ f^{(4)}(\eta,p_{T,\text{cut}}) &= \frac{M\cosh(\eta-y)(Q^2-2p_{T,\text{cut}}^2+2|\mathbf{q}_T|^2)-Q^2\sqrt{M^2\sinh^2(\eta-y)+p_{T,\text{cut}}^2}}{2|\mathbf{q}_T|\left(M^2-p_{T,\text{cut}}^2\right)}\,.\end{aligned} \tag{3.125}$$

Now the question is identifying the integration domain on the $(\eta,\cos\phi)$-plane defined by $\Phi(p,q-p)$ in Eq. (3.124). Considering that $-1\leq\cos\phi\leq 1$, Eq. (3.124) can be written in an more convenient way as

$$\begin{aligned}\Phi(p,q-p) &= \vartheta(\eta-\eta_{\min})\vartheta(\eta_{\max}-\eta)\\ &\times\ \vartheta(\cos\phi-\max[f^{(2)}(\eta,p_{T,\min}),-1])\\ &\times\ \vartheta(\min[f^{(3)}(\eta,\eta_{\min}),f^{(3)}(\eta,\eta_{\max}),f^{(4)}(\eta,p_{T,\min}),1]-\cos\phi)\,.\end{aligned} \tag{3.126}$$

Now we use Eq. (3.115) to change $\cos\phi$ into x. This way, the double integral at the numerator of Eq. (3.113) reads

$$\int_{-\infty}^{\infty}d\eta\int_{-1}^{1}dx\,\Phi(p,q-p)\cdots=\int_{\eta_{\min}}^{\eta_{\max}}d\eta\,\vartheta(x_2(\eta)-x_1(\eta))\int_{x_1(\eta)}^{x_2(\eta)}dx\ldots\,. \tag{3.127}$$

with

$$\begin{aligned}x_1(\eta) &= \max[f^{(2)}(\eta,p_{T,\min}),-1]\\ x_2(\eta) &= \min[f^{(3)}(\eta,\eta_{\min}),f^{(3)}(\eta,\eta_{\max}),f^{(4)}(\eta,p_{T,\min}),1]\,.\end{aligned} \tag{3.128}$$

As an example, Fig. 3.3 shows the integration domain of the numerator of Eq. (3.113) for $p_{T,\min}=20$ GeV and $-\eta_{\min}=\eta_{\max}=2.4$ at $Q=91$ GeV, $|\mathbf{q}_T|=10$ GeV, and $y=1$. The grey band corresponds to the region $-1\leq\cos\phi\leq 1$. The θ-functions in the first line of Eq. (3.126) limits the region to the vertical strip defined by $\eta_{\min}<\eta<\eta_{\max}$ (black vertical lines), the θ-function in the second line defines the region above the red

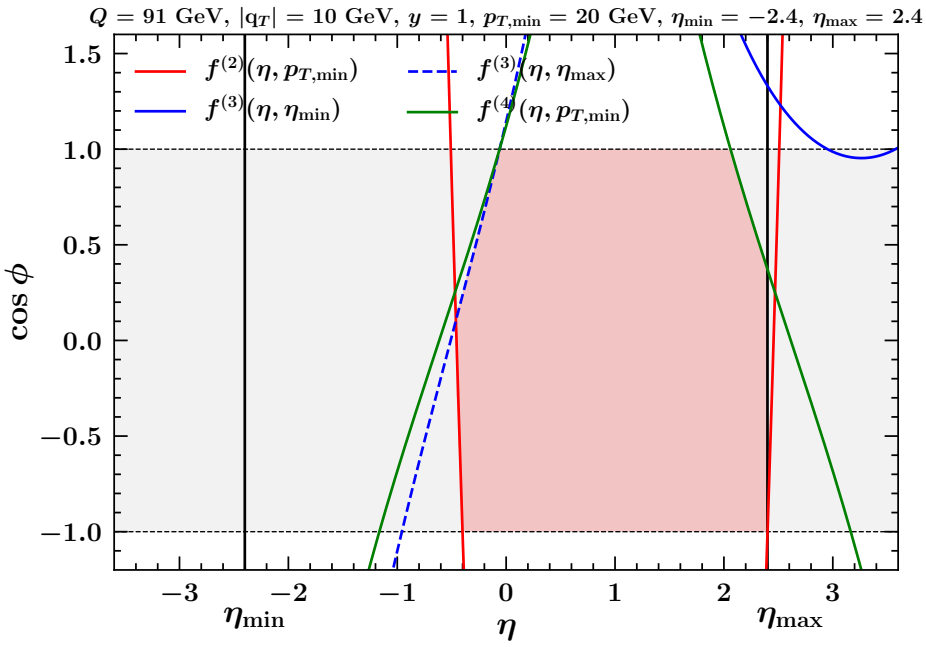


Figure 3.3: The red area indicates the integration domain of the numerator of the phase-space reduction factor Eq. (3.113) for $p_{T,\min} = 20$ GeV and $-\eta_{\min} = \eta_{\max} = 2.4$ at $Q = 91$ GeV, $|\mathbf{q}_T| = 10$ GeV, and $y = 1$.

line, finally the θ-functions in the third line defines the region below the blue and green lines. The intersection of all regions gives the red-shaded area corresponding to the integration domain.

Gathering all pieces, the final expression for the phase-space reduction factor reads

$$\mathcal{P}(q) = \mathcal{P}(Q, y, q_T) = \int_{\eta_{\min}}^{\eta_{\max}} d\eta\, \vartheta(x_2(\eta) - x_1(\eta)) \left[\overline{F}(x_2(\eta), \eta) - \overline{F}(x_1(\eta), \eta)\right] . \quad (3.129)$$

The function $\overline{F}$ is given by the combination

$$\overline{F}(x, \eta) = \frac{3}{4} F(x, \eta) + \frac{1}{4} G(x, \eta) , \quad (3.130)$$

with

$$F(x, \eta) = \frac{1}{4\pi} \frac{Q^2}{E_q^2 - q_T^2} \left\{ \frac{q_T^2 x \sqrt{1 - x^2}}{x^2 q_T^2 - E_q^2} - \frac{E_q}{\sqrt{E_q^2 - q_T^2}} \left[\tan^{-1} \left(\frac{q_T - x E_q}{\sqrt{E_q^2 - q_T^2}\sqrt{1 - x^2}} \right) - \tan^{-1} \left(\frac{q_T + x E_q}{\sqrt{E_q^2 - q_T^2}\sqrt{1 - x^2}} \right) \right] \right\} , \quad (3.131)$$

and

$$G(x,\eta) = \frac{1}{16\pi}\sinh^2(y-\eta)\frac{Q^4}{(E_q^2-q_T^2)^3}\Bigg\{\sqrt{1-x^2}q_T \\ \times\Bigg[\frac{(11E_q^2q_T^2+4q_T^4)x^2+3E_qq_T(9E_q^2+q_T^2)x+(18E_q^4-5E_q^2q_T^2+2q_T^4)}{(xq_T+E_q)^3} \\ +\frac{(11E_q^2q_T^2+4q_T^4)x^2-3E_qq_T(9E_q^2+q_T^2)x+(18E_q^4-5E_q^2q_T^2+2q_T^4)}{(xq_T-E_q)^3}\Bigg] \\ -\frac{6E_q(2E_q^2+3q_T^2)}{\sqrt{E_q^2-q_T^2}}\Bigg[\tan^{-1}\left(\frac{q_T-xE_q}{\sqrt{E_q^2-q_T^2}\sqrt{1-x^2}}\right) \\ -\tan^{-1}\left(\frac{q_T+xE_q}{\sqrt{E_q^2-q_T^2}\sqrt{1-x^2}}\right)\Bigg]\Bigg\}, \tag{3.132}$$

where we have defined $E_q = M\cosh(\eta - y)$ and $q_T = |\mathbf{q}_T|$. Interestingly, in the limit $y = q_T = 0$ and assuming $\eta_{\min} = -\eta_{\max}$, $\mathcal{P}$ can be computed analytically. The result is

$$\mathcal{P}(Q,0,0) = \vartheta(Q-2p_{T,\min})\tanh(\max[\eta_{\max},\overline{\eta}])\left[1-\frac{1}{4\cosh^2(\max[\eta_{\max},\overline{\eta}])}\right], \tag{3.133}$$

with $\overline{\eta}$ defined as

$$\overline{\eta} = \cosh^{-1}\left(\frac{Q}{2p_{T,\min}}\right). \tag{3.134}$$

The relation above can be written more explicitly as

$$\mathcal{P}(Q,0,0) = \begin{cases} 0 & Q < 2p_{T,\min}\,, \\ \left(1-\frac{p_{T,\min}^2}{Q^2}\right)\sqrt{1-\frac{4p_{T,\min}^2}{Q^2}} & 2p_{T,\min} \le Q < 2p_{T,\min}\cosh\eta_{\max}\,, \\ \tanh(\eta_{\max})\left[1-\frac{1}{4\cosh^2(\eta_{\max})}\right] & Q \ge 2p_{T,\min}\cosh\eta_{\max}\,. \end{cases} \tag{3.135}$$

3.A.1 Parity-violating contribution

In the presence of cuts on the final-state leptons and for invariant masses around the Z mass or above, parity-violating effects arise that were neglected in our approach [196]. These contributions stem from interference of the *antisymmetric* contributions to the lepton tensor, proportional to $p_1^\mu p_2^\nu \epsilon_{\mu\nu\rho\sigma}$, and to the hadronic tensor, proportional to

$\epsilon_\perp^{\mu\nu}$ defined as

$$\epsilon_\perp^{\mu\nu} \equiv \epsilon^{\mu\nu\rho\sigma} t_\rho z_\sigma \,, \tag{3.136}$$

where t^μ and z^μ are given in Eq. (3.105). Therefore, the contributions we are after result from the contraction of the following Lorentz structures

$$L_{\rm PV} \equiv p_1^\mu p_2^\nu \epsilon_{\mu\nu\rho\sigma} \epsilon_\perp^{\rho\sigma} = \frac{2|\mathbf{p}_T|^2}{Q} \sinh(y-\eta) \left[M \cosh(y-\eta) - |\mathbf{q}_T| \cos\phi \right] . \tag{3.137}$$

Due to the presence of $\sinh(y-\eta)$, Eq. (3.137) is such that

$$\int_{-\infty}^{\infty} d\eta \, L_{\rm PV} = 0 \,. \tag{3.138}$$

Therefore, for observables inclusive in the lepton phase space, the parity-violating term does not give any contribution. Conversely, the presence of cuts on the final-state leptons may prevent Eq. (3.138) from being satisfied, leaving a residual contribution. In order to quantify this effect, we have taken the same steps performed above to integrate $L_{\rm PV}$ over the fiducial region. It turns out that, for realistic cuts, the numerical size of $\mathcal{P}_{\rm PV}$ relative to the parity-conserving $\mathcal{P}$ is never larger than $\mathcal{O}(10^{-6})$. We conclude that the impact of parity-violating effects in the present analysis is negligible.

3.B Appendix - Ogata quadrature

In this section we limit ourselves to write the formulas for the computation of the unscaled coordinates $z_n^{(\nu)}$ and weights $w_n^{(\nu)}$ required to compute the following integral:

$$\begin{aligned} I_\nu(q_T) = \int_0^\infty db J_\nu(bq_T) f(b) = \frac{1}{q_T} \int_0^\infty d\bar{b} J_\nu(\bar{b}) f\left(\frac{\bar{b}}{q_T}\right) \simeq \frac{1}{q_T} \sum_{n=1}^{\infty} w_n^{(\nu)} f\left(\frac{z_n^{(\nu)}}{q_T}\right) \\ \text{with} \quad \nu = 0, 1, \dots \,, \end{aligned} \tag{3.139}$$

using the Ogata-quadrature algorithm. More details can be found in Ref. [180]. There relevant formulas are:

$$\begin{aligned} z_n^{(\nu)} &= \frac{\pi}{h} \psi\left(\frac{h\xi_{\nu n}}{\pi}\right) , \\ w_n^{(\nu)} &= \pi \frac{Y_\nu(\xi_{\nu n})}{J_{\nu+1}(\xi_{\nu n})} J_\nu(z_n^{(\nu)}) \psi'\left(\frac{h\xi_{\nu n}}{\pi}\right) . \end{aligned} \tag{3.140}$$

where:

- h is a free parameter of the algorithm that has to be typically small (we choose $h = 10^{-3}$),
- $\xi_{\nu n}$ are the zero's of J_ν, *i.e.* $J_\nu(\xi_{\nu n}) = 0 \; \forall \, n$,
- J_ν and Y_ν are the Bessel functions of first and second kind, respectively, of degree ν,
- ψ is the following function:

$$\psi(t) = t \tanh\left(\frac{\pi}{2}\sinh t\right) \tag{3.141}$$

and its derivative:

$$\psi'(t) = \frac{\pi t \cosh t + \sinh(\pi \sinh t)}{1 + \cosh(\pi \sinh t)} \,. \tag{3.142}$$

3.C Appendix - Narrow–width approximation

A possible alternative to the numerical integration in Q when the integration region includes the Z-peak region is the so–called narrow–width approximation (NWA). In the NWA one assumes that the width of the Z boson, Γ_Z, is much smaller than its mass, M_Z. This way one can approximate the peaked behaviour of the couplings $C_q(Q)$ around $Q = M_Z$ with a δ-function, *i.e.*, $C_q(Q) \propto \delta(Q^2 - M_Z^2)$. Therefore, the integration over Q can be done analytically. The exact structure of the electroweak couplings is the following:

$$C_q(Q) = e_q^2 - 2e_q V_q V_e \chi_1(Q) + (V_e^2 + A_e^2)(V_q^2 + A_q^2)\chi_2(Q) \,, \tag{3.143}$$

with:

$$\begin{aligned} \chi_1(Q) &= \frac{1}{4\sin^2\theta_W \cos^2\theta_W} \frac{Q^2(Q^2 - M_Z^2)}{(Q^2 - M_Z^2)^2 + M_Z^2\Gamma_Z^2} \,, \\ \chi_2(Q) &= \frac{1}{16\sin^4\theta_W \cos^4\theta_W} \frac{Q^4}{(Q^2 - M_Z^2)^2 + M_Z^2\Gamma_Z^2} \,. \end{aligned} \tag{3.144}$$

In the limit $\Gamma_Z/M_Z \to 0$, the leading contribution to the coupling in Eq. (3.143) comes from the region $Q \simeq M_Z$ and is that proportional to χ_2:

$$C_q(Q) \simeq (V_e^2 + A_e^2)(V_q^2 + A_q^2)\chi_2(Q) \,, \quad Q \simeq M_Z \,. \tag{3.145}$$

In addition, in this limit one can show that:

$$\frac{1}{(Q^2-M_Z^2)^2+M_Z^2\Gamma_Z^2} \to \frac{\pi}{M_Z\Gamma_Z}\delta(Q^2-M_Z^2) = \frac{\pi}{2M_Z^2\Gamma_Z}\delta(Q-M_Z)\,. \tag{3.146}$$

Therefore, considering that:

$$\Gamma_Z = \frac{\alpha_{\rm em}M_Z}{\sin^2\theta_W\cos^2\theta_W}\,, \tag{3.147}$$

the electroweak couplings in the NWA have the following form:

$$C_q(Q) \simeq \frac{\pi M_Z(V_e^2+A_e^2)(V_q^2+A_q^2)}{32\alpha_{\rm em}\sin^2\theta_W\cos^2\theta_W}\delta(Q-M_Z) = \widetilde{C}_q(Q)\delta(Q-M_Z)\,. \tag{3.148}$$

Therefore, using Eq. (3.145) the integral of the cross section over Q under the condition that $Q_{\rm min} < M_Z < Q_{\rm max}$ has the consequence of fixing the couplings and setting $Q = M_Z$ in the computation. This yields:

$$\int_{Q_{\rm min}}^{Q_{\rm max}} dQ\,\frac{d\sigma}{dQdyd\boldsymbol{q}_T} = \frac{16\pi\alpha_{\rm em}^2 q_T}{9M_Z^3}H(M_Z,M_Z)\sum_q \widetilde{C}_q(M_Z)I_{q\bar{q}}(x_1,x_2,q_T;M_Z,M_Z^2)\,, \tag{3.149}$$

where we are also assuming that $\mu = \sqrt{\zeta} = M_Z$. As a final step, one may want to let the Z boson decay into leptons. At leading order in the EW sector and assuming an equal decay rate for electrons, muons, and tauons, this can be done by multiplying the cross section above by three times the branching ratio for the Z decaying into any pair of leptons, $3{\rm Br}(Z\to\ell^+\ell^-)$.

3.D Appendix - NangaParbat grid points

In this Appendix we report the actual choice of grid points corresponding to the two extractions implemented in `NangaParbat`, PV19 and PV17. In the following, `grid` is a *ThreeDGrid* object. For TMD PDFs (both PV19 and PV17) we chose:

```
grid.Qg = std::vector<double>
{
  1.000000e+00, 1.118034e+00, 1.224745e+00, 1.400000e+00,
  1.581139e+00, 1.788854e+00, 2.000000e+00, 2.236068e+00,
  2.529822e+00, 2.828427e+00, 3.162278e+00, 3.464102e+00,
  4.750000e+00, 5.099020e+00, 6.324555e+00, 7.100000e+00,
  8.000000e+00, 1.000000e+01, 1.118034e+01, 1.224745e+01,
  1.400000e+01, 1.581139e+01, 1.788854e+01, 2.000000e+01,
  2.236068e+01, 2.529822e+01, 2.828427e+01, 3.162278e+01,
```

```
  3.464102e+01, 4.750000e+01, 5.099020e+01, 6.324555e+01,
  7.100000e+01, 8.000000e+01, 1.000000e+02, 1.118034e+02,
  1.224745e+02, 1.400000e+02, 1.581139e+02, 1.788854e+02,
  2.000000e+02
};
grid.xg = std::vector<double>
{
  1.000000e-05, 2.000000e-05, 4.000000e-05, 6.000000e-05,
  8.000000e-05, 1.000000e-04, 2.000000e-04, 4.000000e-04,
  6.000000e-04, 8.000000e-04, 1.000000e-03, 1.500000e-03,
  2.000000e-03, 2.500000e-03, 3.000000e-03, 3.500000e-03,
  4.000000e-03, 4.500000e-03, 5.000000e-03, 5.500000e-03,
  6.000000e-03, 6.500000e-03, 7.000000e-03, 7.500000e-03,
  8.000000e-03, 8.500000e-03, 9.000000e-03, 9.250000e-03,
  9.500000e-03, 9.750000e-03, 1.000000e-02, 1.500000e-02,
  2.000000e-02, 2.500000e-02, 3.000000e-02, 3.500000e-02,
  4.000000e-02, 4.500000e-02, 5.000000e-02, 5.500000e-02,
  6.000000e-02, 6.500000e-02, 7.000000e-02, 7.500000e-02,
  8.000000e-02, 8.500000e-02, 9.000000e-02, 9.250000e-02,
  9.500000e-02, 9.750000e-02, 1.000000e-01, 1.500000e-01,
  2.000000e-01, 2.500000e-01, 3.000000e-01, 3.500000e-01,
  4.000000e-01, 4.500000e-01, 5.000000e-01, 5.500000e-01,
  6.000000e-01, 6.500000e-01, 7.000000e-01, 7.500000e-01,
  8.000000e-01, 8.500000e-01, 9.000000e-01, 9.250000e-01,
  9.500000e-01, 9.750000e-01, 1.000000e+00
};
grid.qToQg = std::vector<double>
{
  0.0001, 0.0010, 0.0025, 0.0050, 0.0075, 0.0100, 0.0200,
  0.0300, 0.0400, 0.0500, 0.0600, 0.0700, 0.0800, 0.0900,
  0.1000, 0.1250, 0.1500, 0.1750, 0.2000, 0.2250, 0.2500,
  0.2750, 0.3000, 0.3500, 0.4000, 0.4500, 0.5000, 0.5500,
  0.6000, 0.6500, 0.7000, 0.8000, 0.9000, 1.0000, 1.1000,
  1.2000, 1.3000, 1.4000, 1.5000, 1.6000, 1.7000, 1.8000,
  1.9000, 2.0010
};
```

while for FFs we have:

```
grid.Qg = std::vector<double>
{
  1.000000e+00, 1.080000e+00, 1.118034e+00, 1.170000e+00,
  1.224745e+00, 1.300000e+00, 1.400000e+00, 1.581139e+00,
  1.788854e+00, 2.000000e+00, 2.236068e+00, 2.529822e+00,
  2.828427e+00, 3.162278e+00, 3.464102e+00, 4.750000e+00,
  5.099020e+00, 6.324555e+00, 7.100000e+00, 8.000000e+00,
  1.000000e+01, 1.118034e+01, 1.224745e+01, 1.400000e+01,
  1.581139e+01, 1.788854e+01, 2.000000e+01, 2.236068e+01,
```

```
  2.529822e+01, 2.828427e+01, 3.162278e+01, 3.464102e+01,
  4.750000e+01, 5.099020e+01, 6.324555e+01, 7.100000e+01,
  8.000000e+01, 1.000000e+02
};
grid.xg = std::vector<double>
{
  1.000000e-01, 1.250000e-01, 1.500000e-01, 1.750000e-01,
  2.000000e-01, 2.250000e-01, 2.500000e-01, 2.750000e-01,
  3.000000e-01, 3.250000e-01, 3.500000e-01, 3.750000e-01,
  4.000000e-01, 4.250000e-01, 4.500000e-01, 4.750000e-01,
  5.000000e-01, 5.250000e-01, 5.500000e-01, 5.750000e-01,
  6.000000e-01, 6.250000e-01, 6.500000e-01, 7.000000e-01,
  7.500000e-01, 8.000000e-01, 8.500000e-01, 9.000000e-01,
  9.500000e-01, 1.000000e+00
};
grid.qToQg = std::vector<double>
{
  0.0001,
  0.0010, 0.0025, 0.0050, 0.0075, 0.0100, 0.0200, 0.0300,
  0.0400, 0.0500, 0.0600, 0.0700, 0.0800, 0.0900, 0.1000,
  0.1100, 0.1200, 0.1300, 0.1400, 0.1500, 0.1600, 0.1700,
  0.1800, 0.1900, 0.2000, 0.2200, 0.2400, 0.2600, 0.2800,
  0.3000, 0.3200, 0.3400, 0.3600, 0.3800, 0.4000, 0.4500,
  0.5000, 0.5500, 0.6000, 0.6500, 0.7000, 0.8000, 0.9000,
  1,
  1.1000, 1.2000, 1.3000, 1.4000, 1.5000, 1.6000, 1.7000,
  1.8000, 1.9000, 2,
  2.1000, 2.2000, 2.3000, 2.4000, 2.5000, 2.6000, 2.7000,
  2.8000, 2.9000, 3,
  3.2000, 3.4000, 3.6000, 3.8000, 4,
  4.2000, 4.4000, 4.6000, 4.8000, 5
};
```

For $F_{\mathrm{UU,T}}$ grids, the standard choice of points in the function `FourDGrid Inter4DGrid (std::string const& pf)` in `inc/NangaParbat/createstructgrid.h` is (in the following `grid` is a *FourDGrid* object):

```
grid.Qg = std::vector<double>
{
  1.000000e+00, 1.080000e+00, 1.118030e+00, 1.170000e+00,
  1.224740e+00, 1.300000e+00, 1.400000e+00, 1.581140e+00,
  1.788850e+00, 2.000000e+00, 2.236070e+00, 2.529820e+00,
  2.828430e+00, 3.162280e+00, 3.464100e+00, 4.750000e+00,
  5.099020e+00, 6.324560e+00, 7.100000e+00, 8.000000e+00,
  1.000000e+01, 1.118030e+01, 1.224750e+01, 1.400000e+01,
  1.581140e+01, 1.788850e+01, 2.000000e+01, 2.236070e+01,
  2.529820e+01, 2.828430e+01, 3.162280e+01, 3.464100e+01,
```

```
  4.750000e+01, 5.099020e+01, 6.324560e+01, 7.100000e+01,
  8.000000e+01, 1.000000e+0
};
grid.xg = std::vector<double>
{
  1.000000e-05, 2.000000e-05, 4.000000e-05, 6.000000e-05,
  8.000000e-05, 1.000000e-04, 2.000000e-04, 4.000000e-04,
  6.000000e-04, 8.000000e-04, 1.000000e-03, 1.500000e-03,
  2.000000e-03, 2.500000e-03, 3.000000e-03, 3.500000e-03,
  4.000000e-03, 4.500000e-03, 5.000000e-03, 5.500000e-03,
  6.000000e-03, 6.500000e-03, 7.000000e-03, 7.500000e-03,
  8.000000e-03, 8.500000e-03, 9.000000e-03, 9.250000e-03,
  9.500000e-03, 9.750000e-03, 1.000000e-02, 1.500000e-02,
  2.000000e-02, 2.500000e-02, 3.000000e-02, 3.500000e-02,
  4.000000e-02, 4.500000e-02, 5.000000e-02, 5.500000e-02,
  6.000000e-02, 6.500000e-02, 7.000000e-02, 7.500000e-02,
  8.000000e-02, 8.500000e-02, 9.000000e-02, 9.250000e-02,
  9.500000e-02, 9.750000e-02, 1.000000e-01, 1.500000e-01,
  2.000000e-01, 2.500000e-01, 3.000000e-01, 3.500000e-01,
  4.000000e-01, 4.500000e-01, 5.000000e-01, 5.500000e-01,
  6.000000e-01, 6.500000e-01, 7.000000e-01, 7.500000e-01,
  8.000000e-01
};
grid.zg = std::vector<double>
{
  0.100, 0.125, 0.150, 0.175,
  0.200, 0.225, 0.250, 0.275,
  0.300, 0.325, 0.350, 0.375,
  0.400, 0.425, 0.450, 0.475,
  0.500, 0.525, 0.550, 0.575,
  0.600, 0.625, 0.650, 0.700,
  0.750, 0.80
};
grid.qToQg =
{
  0.0001, 0.0010, 0.0025, 0.0050, 0.0075, 0.0100, 0.0200,
  0.0300, 0.0400, 0.0500, 0.0600, 0.0700, 0.0800, 0.0900,
  0.1000, 0.1100, 0.1200, 0.1300, 0.1400, 0.1500, 0.1600,
  0.1700, 0.1800, 0.1900, 0.2000, 0.2200, 0.2400, 0.2600,
  0.2800, 0.3000, 0.3200, 0.3400, 0.3600, 0.3800, 0.4000,
  0.4500, 0.5000, 0.5500, 0.6000, 0.6500, 0.7000, 0.8000,
  0.9000, 1
};
```

3.E Appendix - Grid interpolation

In this Appendix we show some results on the comparison between direct calculation of TMDs and interpolation from the corresponding grids. As mentioned in Sec. 3.8, we choose the number of points and the values to put in the grids in order to obtain a satisfactory agreement between direct computation in b_T–space and interpolation of the grid done with the `NangaParbat` interpolator.

In Figs. 3.4–3.7, we show the PV19 and PV17 PDFs, PV17 FF and PV17 unpolarized structure function. To plot the functional forms, we used the central replica of PV19 extraction and the 105th replica of PV17. The motivation of PV17 choice of replica to plot has its roots in Ref. [84], where replica 105 was selected as the most representative of the whole set of replicas. PV17 $f_{\rm NP}$ parameters can be found in Sec. 2.2.2, while the parameter values of PV19 central replica are reported in Tab. 3.1.

Parameter	Central replica value
g_2	0.0379
N_1	0.5188
α	0.2031
σ	0.3733
λ	0.5797
N_{1B}	0.0396
α_B	0.0677
σ_B	0.3631
g_{2B}	0.0112

Table 3.1: Final values of $f_{\rm NP}$ free parameters for (see Eq. (2.42) and (2.43)) of the central replica (replica 0) of the PV19 extraction.

A few comments on the general characteristics of Figs. 3.4–3.7 are in order. In the plots shown in this Appendix, direct computation (in blue) is plotted with a line thicker than the one of grid interpolation in order to make both lines visible when they are superimposed. The graphical representation of the functions in Figs. 3.4–3.7 is obtained connecting with straight lines the points resulting from grid interpolation and computation in b_T–space: in order not to introduce any bias in the comparisons, we did not perform any further intepolation at the plot level to smoothen sharp edges. The lower panels show the ratio between the direct computation and the interpolation.

We stress that the kinematical points shown in the plots do not correspond to the values of $Q, x, z, k_\perp$ or $p_\perp$ for which we computed the grids: in order to test the quality of the multi–dimensional interpolation, we choose on purpose values that do not match the tabulated ones. In fact, we remind that in case the intepolator is called exactly in

the points $(Q, x, k_\perp)$ (or the equivalent for the TMD FF and structure function), the interpolation consists of a mere reading of the grid.

In Fig.3.4 we present the functional form of the PV19 unpolarized TMD PDF extraction for two different choices of x and Q. In particular, at low Q and small x ($Q = 3$ Gev and $x = 0.00015$), the Gaussian and q–Gaussian that compose $f_{\rm NP}$ (see Eq. (2.42)) are clearly visible, while the function becomes smoother as Q and x increase. The agreement between direct computation and f_1 grid interpolation is very good, as shown by the ratios shown in the lower panels. This level of accuracy is stable over a wide kinematic range, *i.e*, for all the $(k_\perp, x, Q)$ points in the range specified in the grid info file , with only a slight degradation at large $k_\perp$, where it follows the same behavior discussed for the PV17 TMD FF below. We note that for PV19 TMDs the relative difference between direct calculation and grid interpolation deteriorates, reaching values up to 5%, around $k_\perp \sim 1.5$ GeV, where the TMD displays a concavity. This deteriorioration is due to the peculiarity of the shape of the TMD, which is very challenging for the interpolation, and, in principle, it could be avoided by increasing significantly the density of the grid points. However, since the problem seems well confined (it is only at $k_\perp \sim 1.5$ GeV), we decided not to increase the number of points, as it would have meant a significant increase also in the size of all grids.

In Fig. 3.5 we show a selection of plots for PV17 TMD PDF. We remind that PV17 extraction included also SIDIS data (while in PV19 we only fitted Drell–Yan data) and the selected functional form for $f_{\rm NP}$ is significantly different from PV19 (see Eq.(2.60)). The difference in the two parameterizations can be clearly seen at low Q and low x: comparing the upper plot of Fig.3.4 with the upper plot of Fig. 3.5, which show PV17 and PV19 f_1 in the same kinematical point, we observe that in PV17 case the PV19 bump is absent and PV17 it has a smoother behaviour overall. Also for PV17 TMD PDF the agreement between direct computation and f_1 grid interpolation is optimal, as shown in the lower panels of Fig. 3.5, where the ratios do not exceed 1%.

In Fig. 3.6 we show direct computation and grid interpolation of the PV17 TMD FF $D_1^{u\to\pi^+}(z, p_\perp; Q^2)$. The agreement between the two different methods is very good, under one or two per thousand, except, in some cases, at large transverse momentum, where the discrepancies between direct calculation and grid interpolation are of the order of a few percent. However, the impact of such discrepancy on the calculation of structure functions, necessary to build experimental observables and obtain predictions, is very small, as shown in Fig. 3.7, and therefore we consider the accuracy of our grids acceptable.

Note that the TMD FF given by the `NangaParbat` interpolator is a function of the transverse momentum of fragmenting parton with respect to the observed hadron $p_\perp$, and not of the transverse momentum of the observed hadron with respect to the fragmenting parton $P_\perp$, as usually indicated in the literature. The reason behind this

choice is that, following Refs. [44, 84, 95, 197], to compute the grids we calculate the Fourier Transform of the TMD FF with respect to the partonic transverse momentum $p_\perp$, even though the unpolarized TMD FF D_1 is understood to have a density interpretation in terms of the hadronic transverse momentum $P_\perp$. We remind that the relation between the two transverse momenta is $P_\perp = -zp_\perp$, which can be obtained from the Lorentz transformation connecting the frames where the fragmenting parton or the produced hadron have no transverse momentum [44, 198].

The plots in Fig. 3.7 are a combined test of both `NangaParbat` interpolation and convolution tools. As discussed in Sec. 3.9.3, the grids for the structure function $F_{\rm UU,T}^{h\to\pi^\pm}$ are obtained through the convolution (see Eq. (3.99)) of two TMD grids (PV17 f_1 and D_1). The interpolation of this four–dimensional grid is compared with the direct calculation of $F_{\rm UU,T}^{h\to\pi^\pm}$ and the results are illustrated in Fig. 3.7. We can see that $F_{\rm UU,T}^{h\to\pi^\pm}$ is very well encoded in the grids we produced, as the discrepancy between the two lines in Fig. 3.7 is of the order of per thousand at $Q \sim 1$ GeV. At higher values of Q the relative difference between direct calculation and grid interpolation degrades slightly, settling around 1%. As this is an interpolation on four dimensions, we still consider this to be a good result. This discrepancies could be cured by increasing the number of points present in the grids, but that would be at the expense of the size of the grid files, which we managed to keep reasonable and suited for sharing. For analyses that require an accuracy higher than 1% or 2%, we recommend using the convolution tool illustrated in Sec. 3.9.3, which convolutes two TMD grids and gives results where the discrepancies with respect to direct calculation are below 1%. $F_{\rm UU,T}^{h\to\pi^\pm}$ is here plotted as a function of $(x, z, q_T; Q^2)$: we remind that q_T is the transverse momentum of the exchanged photon in the frame where the target hadron and the observed hadron are collinear, and the relation with the transverse momentum P_{hT} is $q_T = -P_{hT}/z$.

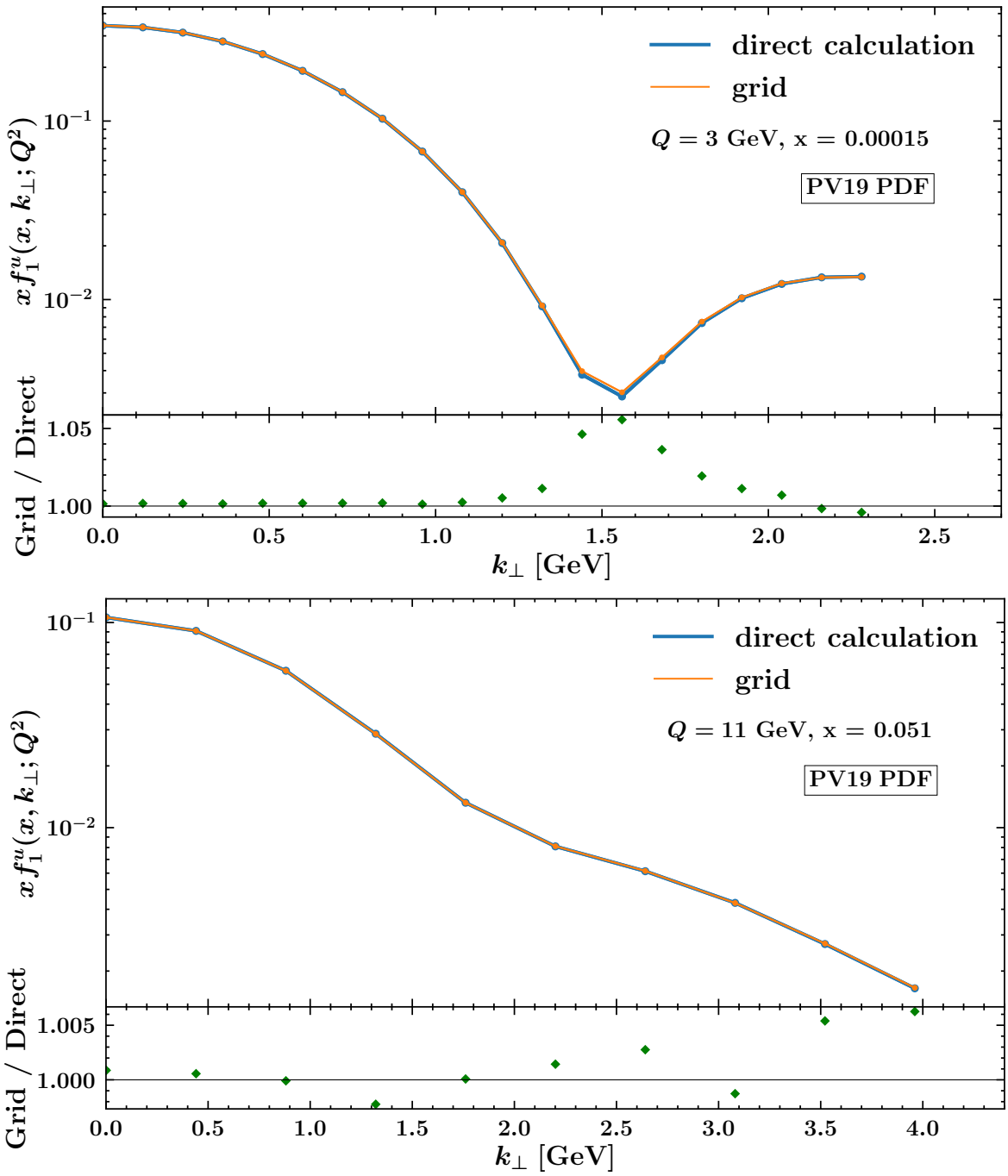


Figure 3.4: The unpolarized TMD PDF $f_1^u(x, k_\perp; Q^2)$ of the PV19 extraction[95], shown here multiplied by x as is often done in the literature.

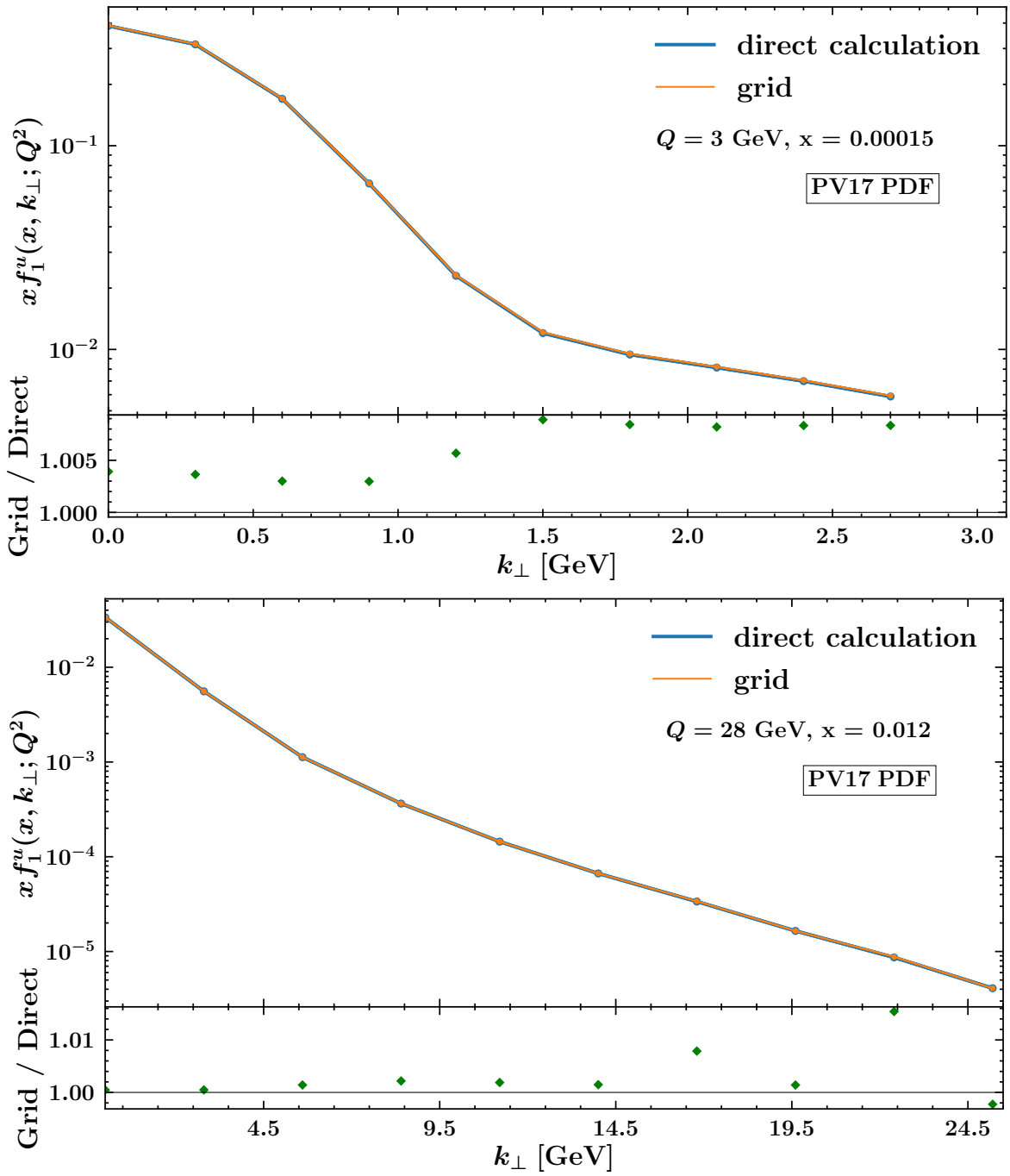


Figure 3.5: The unpolarized TMD PDF $f_1^u(x, k_\perp; Q^2)$ of the PV17 extraction[84], shown here multiplied by x as is often done in the literature.

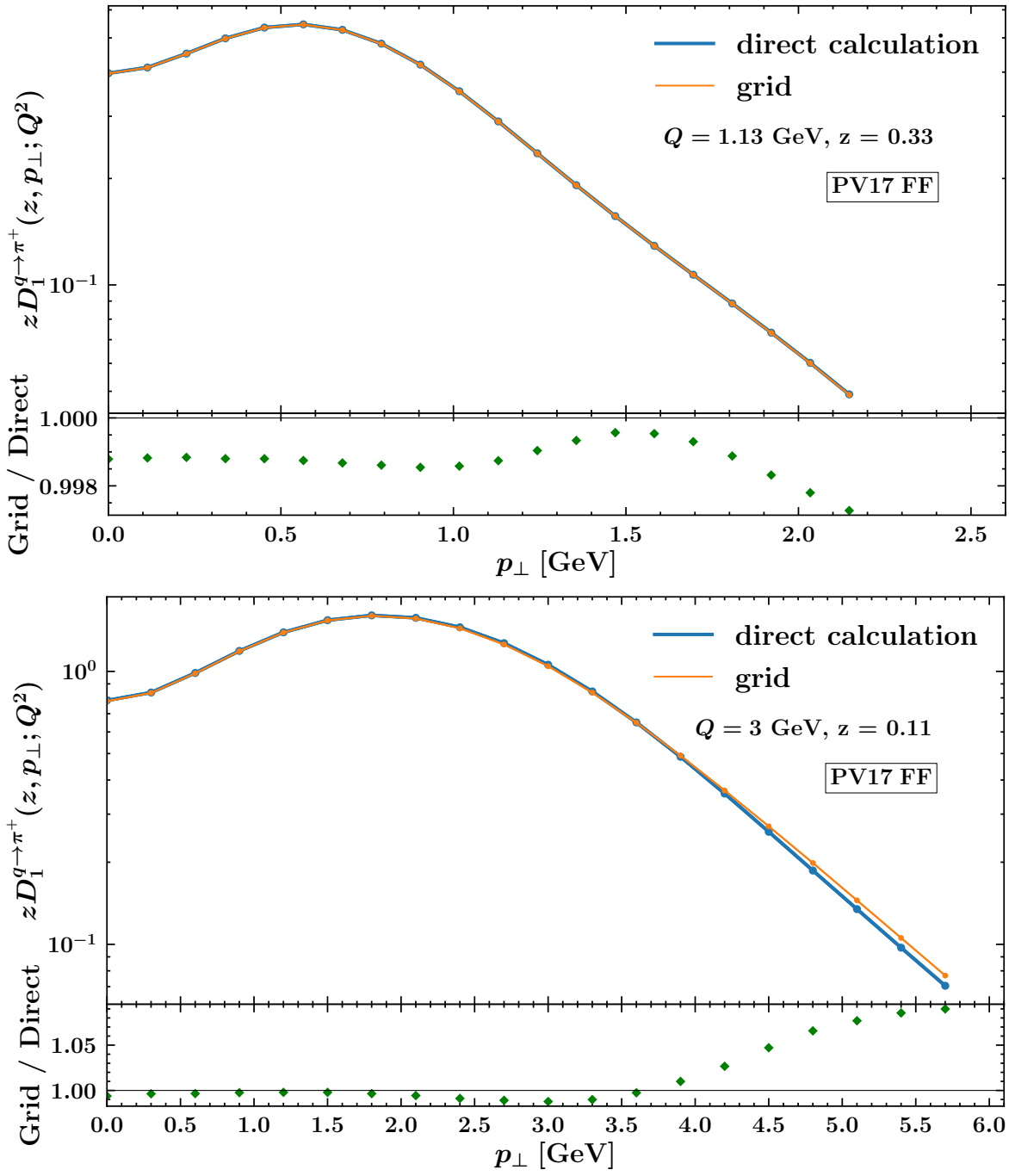


Figure 3.6: PV17 TMD FF $D_1^{q\to\pi^+}(z, p_\perp; Q^2)$, plotted here for the up quark ($q = u$) and for two different values of Q and z.

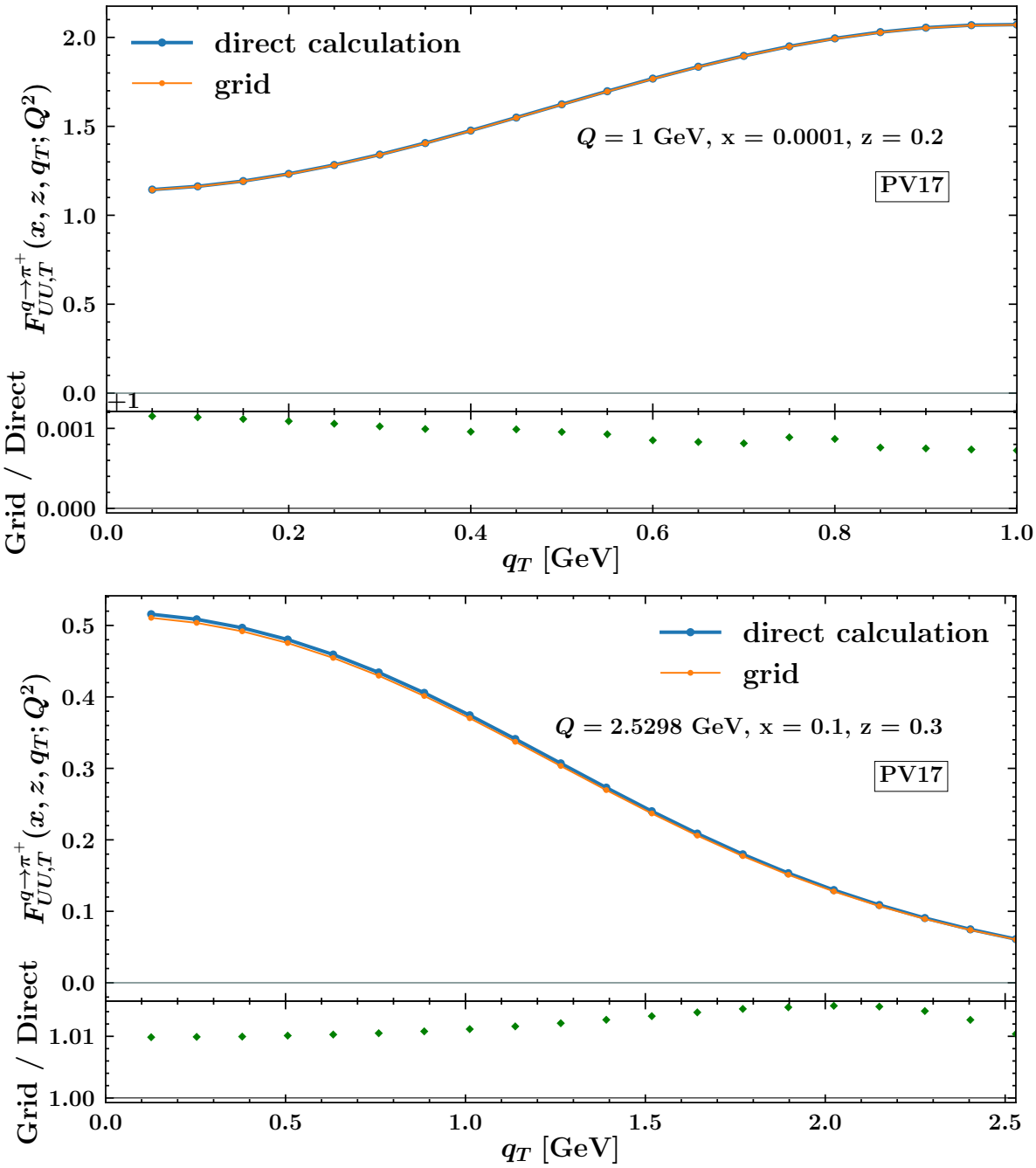


Figure 3.7: Structure function $F_{\mathrm{UU,T}}^{h\to\pi^+}$. The agreement between direct computation and the results obtained through the convolution (see Eq. (3.99)) of PV17 f_1 and D_1 grids is very good at $Q = 1$ and is under 2% for higher values of Q.

Chapter 4

TMD PDF extraction at N^{3}LL

In this chapter, we report the results of our most recent DY fit, which in this work we will refer to as the PV19 fit. In particular, in Sec. 4.1, we discuss the datasets that we selected for our analysis, in Sec. 4.2, the uncertainties treatment is briefly illustrated; in Sec. 4.3, we present the quality of the fit at N^{3}LL, the best accuracy we can presently reach, and we discuss the TMDs extracted from the nominal fit and their parameters; in Sec. 4.4, we discuss the convergence of the perturbative corrections; in Sec. 4.5, we focus on the x dependence of the TMDs and we argue that it is mostly constrained by the y–differential ATLAS cross sections; in Sec. 4.6, we assess the range of validity of TMD factorization by considering the fit quality as a function of the cut on q_T/Q.

4.1 Drell–Yan experimental data

We considered q_T distributions in DY production from many different datasets. Some of these were already considered in the analysis of Ref. [84]: the low–energy fixed target experiments E605 [199] and E288 [200],[1] and the Tevatron experiments CDF Run I [201], Run II [202], D0 Run I [203] and Run II [204]. We refer the reader to Ref. [84] for more details about these datasets. In the PV19 fit, we included the following new datasets:

- $Z \to \mu^+\mu^-$ distribution from D0 Run II [205],
- forward Z-production data from the LHCb experiment at 7 [205], 8 [206], and 13 [207] TeV,
- Z-production data from the CMS experiment at 7 [208] and 8 [209] TeV,

[1] The fixed–target Drell–Yan data from the E288 and E605 experiments used a copper target. In our analysis, such target is considered perfectly isoscalar.

- Z-production data differential in rapidity from the ATLAS experiment at 7 [208] and 8 [210] TeV,
- off-peak (low- and high-mass) DY data from the ATLAS experiment at 8 TeV [210],
- preliminary Z-production data from the STAR experiment at 510 GeV [211].[2]

We did not include measurements from the PHENIX experiment at the center–of–mass energy of 200 GeV [212] because, due to the cut on q_T/Q discussed below, only two data points from this dataset would be included in the fit.

In Tab. 4.1 all the datasets we included in our analysis are shown. For visualization purposes, in Fig. 4.1 we show the kinematic coverage of each dataset in the x_1 vs. x_2 plane. The shaded areas are determined considering the corresponding ranges in Q and y, and the center–of–mass energy $\sqrt{s}$, according to $x_{1,2} = Qe^{\pm y}/\sqrt{s}$. It should be kept in mind that Fig. 4.1 only provides an approximated view of the real coverage, strictly true only at tree level. The reason is that x_1 and x_2 are just the lower bounds of convolution integrals (see, *e.g.*, Eq. (2.18)). Therefore, the effective region of sensitivity actually extends between $x_{1,2}$ and 1.

E605 and E288, being fixed target experiments with fixed rapidity, are just lines in this pictorial representation, while all other datasets are shown as rectangles with their proper range in Q and y. As expected, the lower–energy experiments (E605, E288, and STAR) are placed in the large–x region ($x \gtrsim 0.1$). Particularly important are the new STAR measurements that cover a kinematic region that is scarcely populated.

The Tevatron experiments, CDF and D0, cover a particularly wide kinematic region at intermediate values of x. These experiments (except D0 Run II with muons) provide data extrapolated over the full range in rapidity y, thus extending across the full available phase space.

Finally, the LHC experiments (LHCb, CMS, and ATLAS) are placed at lower values of x. The LHCb datasets are peculiar within respect to the others, in a region in which x_1 is particularly small and x_2 particularly large: this is due to the fact that the data is taken in the forward region, $2 < y < 4.5$. The ATLAS datasets are binned in rapidity (in Fig. 4.1 y–bins are the boxes on the upper–right–to–lower–left diagonal) and thus are expected to be particularly sensitive to the x dependence of the TMDs.

Since our analysis is based on the TMD factorization formula in Eq. (2.10), that is expected to hold for $q_T \ll Q$, we aim to describe only data at small q_T. Hence, some rule to exclude measurements with large q_T is needed: we chose to impose a cut on q_T, requiring $q_T/Q < 0.2$. We remark that such cut is imposed *a priori* and both the value (0.2) and the variable (q_T/Q) on which the cut is imposed are arbitrary.

[2] We thank the STAR Collaboration for providing us with the data, which at the time of publication of the PV19 fit [95] were preliminary.

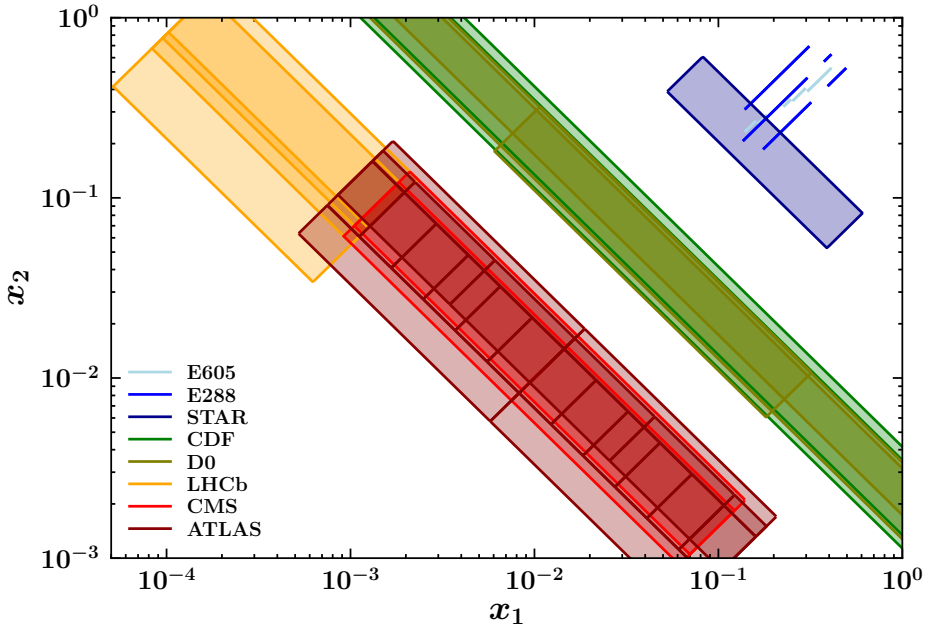


Figure 4.1: Kinematic coverage on the x_1 vs. x_2 plane of the datasets included in the present analysis.

Different choices can be made, as for example in Ref. [85]. The impact of the choice of such cut on fit results is discussed in Sec. 4.6. Since the measurements are delivered in transverse-momentum bins $[q_{T,\min}: q_{T,\max}]$ integrated over some range in invariant mass $[Q_{\min}: Q_{\max}]$, the cut is conservatively imposed on the ratio $q_{T,\max}/Q_{\min}$. The second column in Tab. 4.1 reports the number of data points ($N_{\rm dat}$) for each dataset that pass this cut: the total number of points included in our analysis is 353.

An important feature of all the new datasets listed above is that the cross sections are given within a certain fiducial region. In particular, in order to be properly considered in the phenomenological analysis, experimental data, due to the construction of the detectors and their acceptance, impose kinematic cuts on transverse momentum $p_{T\ell}$ and pseudo–rapidity η_ℓ of the final–state leptons. The values of the cuts for each dataset are shown in the next–to–last column of Tab. 4.1. Our predictions are corrected by means of the phase–space reduction factor $\mathcal{P}$ introduced in Eq. (2.10), which takes into account these cuts, as explained in Appendix 3.A.

As evident from the "Observable" column of Tab. 4.1, experimental cross sections are released in different forms. In addition, some of them are normalized to the total (fiducial) cross section while others are not. In our analysis, we expressed all the absolute cross sections in terms of the observable given in Eq. (2.10) (details on the transformations between different observables can be found in Ref. [84]).

When necessary, the total cross section σ required to normalize the differential

cross sections is computed using DYNNLO [213, 214] with the MMHT2014 collinear PDF sets [126], taking into account the selection cuts and consistently with the perturbative order of the differential cross section. More precisely, the total cross section is computed at LO for NLL accuracy, at NLO for NLL′ and NNLL, and at NNLO for NNLL′ and N^3LL. The values of the total cross sections at different orders are reported in Tab. 4.2.

We stress that in this analysis no additional normalizations have been applied, with the consequence that both the shape and the normalization of the experimental distributions have an impact on the fit.

Experiment	$N_{\rm dat}$	Observable	$\sqrt{s}$ [GeV]	Q [GeV]	y or x_F	Lepton cuts	Ref.
E605	50	$Ed^3\sigma/d^3q$	38.8	7 - 18	$x_F = 0.1$	-	[199]
E288 200 GeV	30	$Ed^3\sigma/d^3q$	19.4	4 - 9	$y = 0.40$	-	[200]
E288 300 GeV	39	$Ed^3\sigma/d^3q$	23.8	4 - 12	$y = 0.21$	-	[200]
E288 400 GeV	61	$Ed^3\sigma/d^3q$	27.4	5 - 14	$y = 0.03$	-	[200]
STAR 510	7	$d\sigma/dq_T$	510	73 - 114	$\lvert y\rvert < 1$	$p_{T\ell} > 25$ GeV $\lvert\eta_\ell\rvert < 1$	[211]
CDF Run I	25	$d\sigma/dq_T$	1800	66 - 116	Inclusive	-	[201]
CDF Run II	26	$d\sigma/dq_T$	1960	66 - 116	Inclusive	-	[202]
D0 Run I	12	$d\sigma/dq_T$	1800	75 - 105	Inclusive	-	[203]
D0 Run II	5	$(1/\sigma)d\sigma/dq_T$	1960	70 - 110	Inclusive	-	[204]
D0 Run II (μ)	3	$(1/\sigma)d\sigma/dq_T$	1960	65 - 115	$\lvert y\rvert < 1.7$	$p_{T\ell} > 15$ GeV $\lvert\eta_\ell\rvert < 1.7$	[205]
LHCb 7 TeV	7	$d\sigma/dq_T$	7000	60 - 120	$2 < y < 4.5$	$p_{T\ell} > 20$ GeV $2 < \eta_\ell < 4.5$	[206]
LHCb 8 TeV	7	$d\sigma/dq_T$	8000	60 - 120	$2 < y < 4.5$	$p_{T\ell} > 20$ GeV $2 < \eta_\ell < 4.5$	[207]
LHCb 13 TeV	7	$d\sigma/dq_T$	13000	60 - 120	$2 < y < 4.5$	$p_{T\ell} > 20$ GeV $2 < \eta_\ell < 4.5$	[215]
CMS 7 TeV	4	$(1/\sigma)d\sigma/dq_T$	7000	60 - 120	$\lvert y\rvert < 2.1$	$p_{T\ell} > 20$ GeV $\lvert\eta_\ell\rvert < 2.1$	[208]
CMS 8 TeV	4	$(1/\sigma)d\sigma/dq_T$	8000	60 - 120	$\lvert y\rvert < 2.1$	$p_{T\ell} > 15$ GeV $\lvert\eta_\ell\rvert < 2.1$	[209]
ATLAS 7 TeV	6 6 6	$(1/\sigma)d\sigma/dq_T$	7000	66 - 116	$\lvert y\rvert < 1$ $1 < \lvert y\rvert < 2$ $2 < \lvert y\rvert < 2.4$	$p_{T\ell} > 20$ GeV $\lvert\eta_\ell\rvert < 2.4$	[216]
ATLAS 8 TeV on-peak	6 6 6 6 6 6	$(1/\sigma)d\sigma/dq_T$	8000	66 - 116	$\lvert y\rvert < 0.4$ $0.4 < \lvert y\rvert < 0.8$ $0.8 < \lvert y\rvert < 1.2$ $1.2 < \lvert y\rvert < 1.6$ $1.6 < \lvert y\rvert < 2$ $2 < \lvert y\rvert < 2.4$	$p_{T\ell} > 20$ GeV $\lvert\eta_\ell\rvert < 2.4$	[210]
ATLAS 8 TeV off-peak	4 8	$(1/\sigma)d\sigma/dq_T$	8000	46 - 66 116 - 150	$\lvert y\rvert < 2.4$	$p_{T\ell} > 20$ GeV $\lvert\eta_\ell\rvert < 2.4$	[210]
Total	353	-	-	-	-	-	-

Table 4.1: Datasets included in this analysis. For each dataset, the table includes information on: the number of data points ($N_{\rm dat}$) that survive the nominal cut on q_T/Q, the observable delivered, the center of mass energy $\sqrt{s}$, the range(s) in invariant mass Q, the angular variable (either y or x_F), possible cuts on the single final-state leptons, and the public reference (when available). The total number of data points amounts to 353. Note that for E605 and E288 400 GeV we have excluded the bin in Q containing the Υ resonance ($Q \simeq 9.5$ GeV).

Experiment		LO [pb]	NLO [pb]	NNLO [pb]
D0 Run II		170.332	242.077	253.573
D0 Run II (μ)		100.765	119.002	124.675
CMS 7 TeV		291.977	384.569	398.853
CMS 8 TeV		340.132	456.337	473.411
ATLAS 7 TeV	$\lvert y\rvert < 1$	196.457	251.296	253.781
	$1 < \lvert y\rvert < 2$	135.511	181.267	181.466
	$2 < \lvert y\rvert < 2.4$	12.568	17.091	17.104
ATLAS 8 TeV on-peak	$\lvert y\rvert < 0.4$	89.531	113.650	116.766
	$0.4 < \lvert y\rvert < 0.8$	89.120	112.853	115.738
	$0.8 < \lvert y\rvert < 1.2$	85.499	109.800	112.457
	$1.2 < \lvert y\rvert < 1.6$	69.018	91.884	95.187
	$1.6 < \lvert y\rvert < 2$	43.597	59.114	62.127
	$2 < \lvert y\rvert < 2.4$	14.398	19.574	20.937
ATLAS 8 TeV off-peak	46 GeV $< Q <$ 66 GeV	15.199	14.449	14.368
	116 GeV $< Q <$ 150 GeV	3.805	5.317	5.521

Table 4.2: Total (fiducial) cross sections computed with `DYNNLO` [213, 214] using the central member of the `MMHT2014` collinear PDF sets [126] and required for the computation of the normalized differential cross sections at the different perturbative orders.

4.2 Experimental uncertainties

One of the characteristics of the PV19 fit is the treatment of uncertainties, as discussed in Ch. 3, Sec. 3.6.

Most of the considered experimental datasets are released with a set of uncorrelated and correlated uncertainties. We remark that such nature of the uncertainties, as it is related to how the particular experiment was devised and how the specific dataset was extracted, has to be specified by experimental collaborations with the release of each dataset.

As already pointed out in Ref. [79], a proper treatment of the experimental uncertainties is crucial to achieve a reliable extraction of TMDs, and particular care has to be taken with normalization errors, in order to avoid the D'Agostini bias [185, 186] (see Sec. 3.6.2). Different prescriptions have been devised to avoid this problem [188]: in this analysis we adopt the so–called iterative t_0-prescription [187].

This topic, as well as the actual implementation of the χ^2, is extensively discussed in Ch. 3, Sec. 3.6. It is useful, though, to remind the reader the concept of *shifted prediction*, which is what will be plotted in Sec. 4.3. In the presence of correlated uncertainties, the χ^2 can be written as [188]

$$\chi^2 = \chi^2_D + \chi^2_\lambda \,, \tag{4.1}$$

where χ^2_D has an uncorrelated structure (diagonal) while χ^2_λ is a *penalty term* related to the presence of correlations [79](see Sec. 3.6.3).

For the computation of χ^2_D, theoretical predictions are properly shifted to take into account the effect of the correlated uncertainties. In fact, shifted predictions are a better representation of the fit for visual comparisons to experimental data. Therefore, in the following it is understood that all plots will display shifted predictions.

Another important aspect, that was not taken into account in Ref. [84], is the use of collinear PDFs to estimate the uncertainties due to the collinear theoretical terms (which are not fitted in this analysis). In previous studies, such as Ref. [84], experimental errors of the selected datasets used for TMD extraction were of the order of %, thus bigger than uncertainties coming from other sources (*i.e.* PDF uncertainties or theoretical uncertainties). In this analysis, though, the inclusion of LHC data, with their very small errors (ATLAS data have very small uncertainties, much lower than 1%), required a more careful study of other types of uncertainties. We noticed, for example, that uncertainties coming from the chosen collinear PDF set were comparable with ATLAS experimental errors, and thus we decided to include them in our analysis.

In order to extract $f_{\rm NP}$ defined in Eq. (2.39), it is necessary to assume a given set of collinear PDFs (in the PV19 analysis, we used `MMHT2014` PDF sets).

PDF uncertainties indirectly reflect the experimental uncertainty of the dataset used for their extraction and have to be included in the calculation of the χ^2.

To do so, we computed the PDF errors, using the Hessian method [217], as relative to the central value. The advantage of computing relative uncertainties is that of minimizing the dependence on the nonperturbative function $f_{\rm NP}$. The impact of PDF uncertainties on the observable X (the Drell–Yan cross section in our case) was calculated using the Hessian formula:

$$\Delta X = \frac{1}{2}\left(\sum_{i=1}^{N_p}\left[X(S_i^+) - X(S_i^-)\right]^2\right)^{\frac{1}{2}} \tag{4.2}$$

where N_p is the number of hessian couples present in the PDF set and S_i^+, S_i^- are respectively the upper and lower member of the hessian couple. The authors of the hessian PDF set have to specify which members are to be considered as S_i^+ and which ones as S_i^-. In order to compute the cross sections $X(S_i^+), X(S_i^-)$ some approximations have been made (see Ch. 3, Sec. 3.4.1). We also point out that the calculation of PDF uncertainties does include the uncertainty on the total cross sections when normalized distributions are considered. PDF uncertainties calculated in this way are included in the experimental covariance matrix as uncorrelated uncertainties.

The computation of the experimental covariance matrix is done in `NangaParbat`, as explained in Ch. 3, Sec. 3.6.1- 3.6.2.

The propagation of the resulting experimental uncertainty into the fitted TMDs is achieved through a Monte Carlo method. Specifically, starting from the actual data, we generate pseudo–data, shifting each experimental point by a random amount calculated with Gaussian distribution. The standard deviation of this Gaussian distribution is obtained taking into account all the uncertainties.

In this way, we obtain $N_{\rm rep}$ ($\gtrsim 200$) replicas of the original dataset and then perform a fit on each single replica. The number of replicas is chosen so that the mean and standard deviation of the set of replicas accurately reproduces the original data points. In particular, in we choose to select 200 replicas as in the PV17 analysis, since we found that such number of replicas is sufficient for the purpose. The resulting ensemble of distributions can be used to compute central values and uncertainties as averages and correlations, respectively.

4.3 Fit quality

In this section, we discuss the quality of the PV19 nominal fit at N^{3}LL [95]. In order to show our results and to quantify the agreement of our predictions with experimental

data, we chose to evaluate the χ^2s using the mean of the TMDs extracted from the Monte Carlo replicas of the data. Denoting the Monte Carlo ensemble of TMDs with $\{\tilde{f}_1^{q,[k]}\}$, $k = 1, \ldots, N_{\rm rep}$ ($N_{\rm rep}$ being the number of replicas), the *mean* is defined as

$$\tilde{f}_1^q(x, b_T; \mu, \zeta) = \frac{1}{N_{\rm rep}} \sum_{k=1}^{N_{\rm rep}} \tilde{f}_1^{q,[k]}(x, b_T; \mu, \zeta)\,. \tag{4.3}$$

Other choices are possible, such as the median or the mode of the ensemble. In fact, only the full ensemble of replicas carries the full statistical information.

Tab. 4.3 reports the breakdown of the χ^2s normalized to the number of data points, $N_{\rm dat}$, for each dataset. The uncorrelated (χ^2_D) and the correlated (χ^2_λ) contributions to the total χ^2 (see Eq. (4.1)) are also reported. The global χ^2 is shown at the bottom of the table.

The value of the global χ^2 is very close to one (1.02), indicating that the fit is able to describe measurements over a wide energy range, from the low–energy fixed–target datasets to the LHC ones. It is important to stress that a substantial contribution to the global χ^2 is given by the correlated penalty term, $\chi^2_\lambda/N_{\rm dat} = 0.14$. This highlights the importance of a correct treatment of the correlated uncertainties. More specifically, the systematic shifts induced by correlations are often large, indicating that the fit does need to adjust the predictions within the experimentally correlated ranges.

Concerning the single experiments, we observe that the low–energy data (E605, E288, and STAR) have generally lower χ^2s than the Tevatron (CDF and D0) and LHC (LHCb, CMS, and ATLAS) high–energy data. This is mostly due to the fact that the experimental uncertainties of the former are typically larger than the latter. In particular, the low–energy data are affected by large normalization (correlated) uncertainties (up to 15% in some cases). Consequently, the relative importance of the correlated contribution χ^2_λ to the total χ^2 is generally larger for the low–energy datasets than for the high–energy ones.

It is interesting to comment on the quality of the fit of the new datasets from RHIC and the LHC that were not included in the analysis of Ref. [84]. The preliminary measurements from STAR have a χ^2 equal to 0.836. This is particularly encouraging because, as clear from Fig. 4.1, this dataset covers a scarcely populated kinematic region and shows no tension with other data.

Also the LHC datasets extend the kinematic coverage of the DY data considered in Ref. [84]. These measurements are particularly precise and thus very effective in constraining TMDs. We observe that the LHCb datasets are very nicely described with χ^2s that never exceed 1.3. The CMS data, despite having slightly larger χ^2, are also well described. The CMS 7 TeV experiment is the only one that does not provide information on correlated errors, and this reflects in the $\chi^2_\lambda/N_{\rm dat} = 0$ in Tab. 4.3. The

Experiment		$\chi^2_D/N_{\rm dat}$	$\chi^2_\lambda/N_{\rm dat}$	$\chi^2/N_{\rm dat}$
E605	7 GeV $< Q <$ 8 GeV	0.419	0.068	0.487
	8 GeV $< Q <$ 9 GeV	0.995	0.034	1.029
	10.5 GeV $< Q <$ 11.5 GeV	0.191	0.137	0.328
	11.5 GeV $< Q <$ 13.5 GeV	0.491	0.284	0.775
	13.5 GeV $< Q <$ 18 GeV	0.491	0.385	0.877
E288 200 GeV	4 GeV $< Q <$ 5 GeV	0.213	0.649	0.862
	5 GeV $< Q <$ 6 GeV	0.673	0.292	0.965
	6 GeV $< Q <$ 7 GeV	0.133	0.141	0.275
	7 GeV $< Q <$ 8 GeV	0.254	0.014	0.268
	8 GeV $< Q <$ 9 GeV	0.652	0.024	0.676
E288 300 GeV	4 GeV $< Q <$ 5 GeV	0.231	0.555	0.785
	5 GeV $< Q <$ 6 GeV	0.502	0.204	0.706
	6 GeV $< Q <$ 7 GeV	0.315	0.063	0.378
	7 GeV $< Q <$ 8 GeV	0.056	0.030	0.086
	8 GeV $< Q <$ 9 GeV	0.530	0.017	0.547
	11 GeV $< Q <$ 12 GeV	1.047	0.167	1.215
E288 400 GeV	5 GeV $< Q <$ 6 GeV	0.312	0.065	0.377
	6 GeV $< Q <$ 7 GeV	0.100	0.005	0.105
	7 GeV $< Q <$ 8 GeV	0.018	0.011	0.029
	8 GeV $< Q <$ 9 GeV	0.437	0.039	0.477
	11 GeV $< Q <$ 12 GeV	0.637	0.036	0.673
	12 GeV $< Q <$ 13 GeV	0.788	0.028	0.816
	13 GeV $< Q <$ 14 GeV	1.064	0.044	1.107
STAR		0.782	0.054	0.836
CDF Run I		0.480	0.058	0.538
CDF Run II		0.959	0.001	0.959
D0 Run I		0.711	0.043	0.753
D0 Run II		1.325	0.612	1.937
D0 Run II (μ)		3.196	0.023	3.218
LHCb 7 TeV		1.069	0.194	1.263
LHCb 8 TeV		0.460	0.075	0.535
LHCb 13 TeV		0.735	0.020	0.755
CMS 7 TeV		2.131	0.000	2.131
CMS 8 TeV		1.405	0.007	1.412
ATLAS 7 TeV	$0 < \lvert y\rvert < 1$	2.581	0.028	2.609
	$1 < \lvert y\rvert < 2$	4.333	1.032	5.365
	$2 < \lvert y\rvert < 2.4$	3.561	0.378	3.939
ATLAS 8 TeV on-peak	$0 < \lvert y\rvert < 0.4$	1.924	0.337	2.262
	$0.4 < \lvert y\rvert < 0.8$	2.342	0.247	2.590
	$0.8 < \lvert y\rvert < 1.2$	0.917	0.061	0.978
	$1.2 < \lvert y\rvert < 1.6$	0.912	0.095	1.006
	$1.6 < \lvert y\rvert < 2$	0.721	0.092	0.814
	$2 < \lvert y\rvert < 2.4$	0.932	0.348	1.280
ATLAS 8 TeV off-peak	46 GeV $< Q <$ 66 GeV	2.138	0.745	2.883
	116 GeV $< Q <$ 150 GeV	0.501	0.003	0.504
Global		**0.88**	**0.14**	**1.02**

Table 4.3: The $\chi^2/N_{\rm dat}$ using the mean replica in Eq. (4.3). $N_{\rm dat}$ in each case is listed in Tab. 4.1. The uncorrelated (χ^2_D) and correlated (χ^2_λ) contributions and their sum χ^2 are shown (see Eq. (4.1)).

two CMS datasets provide only eight points in total and thus their impact on the fit is modest. The ATLAS datasets, among the LHC ones, are by far the most abundant. We observe that the ATLAS 8 TeV datasets are well described, except for the first two low–rapidity bins. The 7 TeV ones present larger values of χ^2, above 2. Given the extremely high precision of these datasets, even small effects (e.g., power corrections) could give a significant contribution to χ^2 in these conditions. We consider it already a success to obtain a value of χ^2 for these datasets that does not affect too much the global χ^2. We note that a key feature of these datasets (except the off-peak ones) is that they are differential in the vector-boson rapidity y. As we will see in Sec. 4.5, the x dependence of $f_{\rm NP}$ plays a crucial role in improving the χ^2.

In order to provide a visual assessment of the fit quality, Fig. 4.2 displays the data/theory comparison for a representative selection of datasets. We remind the reader that in each plot theoretical predictions are appropriately shifted to account for correlated uncertainties [79], while the experimental error bars are given by the sum in quadrature of the uncorrelated uncertainties. The upper panel of each plot shows the absolute q_T distribution, while the lower panel shows the ratio to data. The plots in the upper row of Fig. 4.2 refer to one invariant–mass bin of E288 and D0 Run I already considered in Ref. [84]. The remaining plots refer to some of the new datasets, namely STAR, LHCb 7 TeV, ATLAS 7 TeV on–peak at $1 < |y| < 2$, ATLAS 8 TeV on–peak at $0.8 < |y| < 1.2$ and $1.2 < |y| < 1.6$ and ATLAS 8 TeV off-peak at 116 GeV $< Q <$ 150 GeV. As expected, there is a very good agreement between data and theory, for both the old and the new datasets. Finally, it is interesting to observe that the uncertainties of E288, D0, STAR and LHCb in Fig. 4.2 are larger than the ATLAS ones in the bottom rows. This is due to the fact that the ATLAS distributions are normalized to the total cross section leading to a cancellation of some uncertainties, such as those due to luminosity and collinear PDFs.

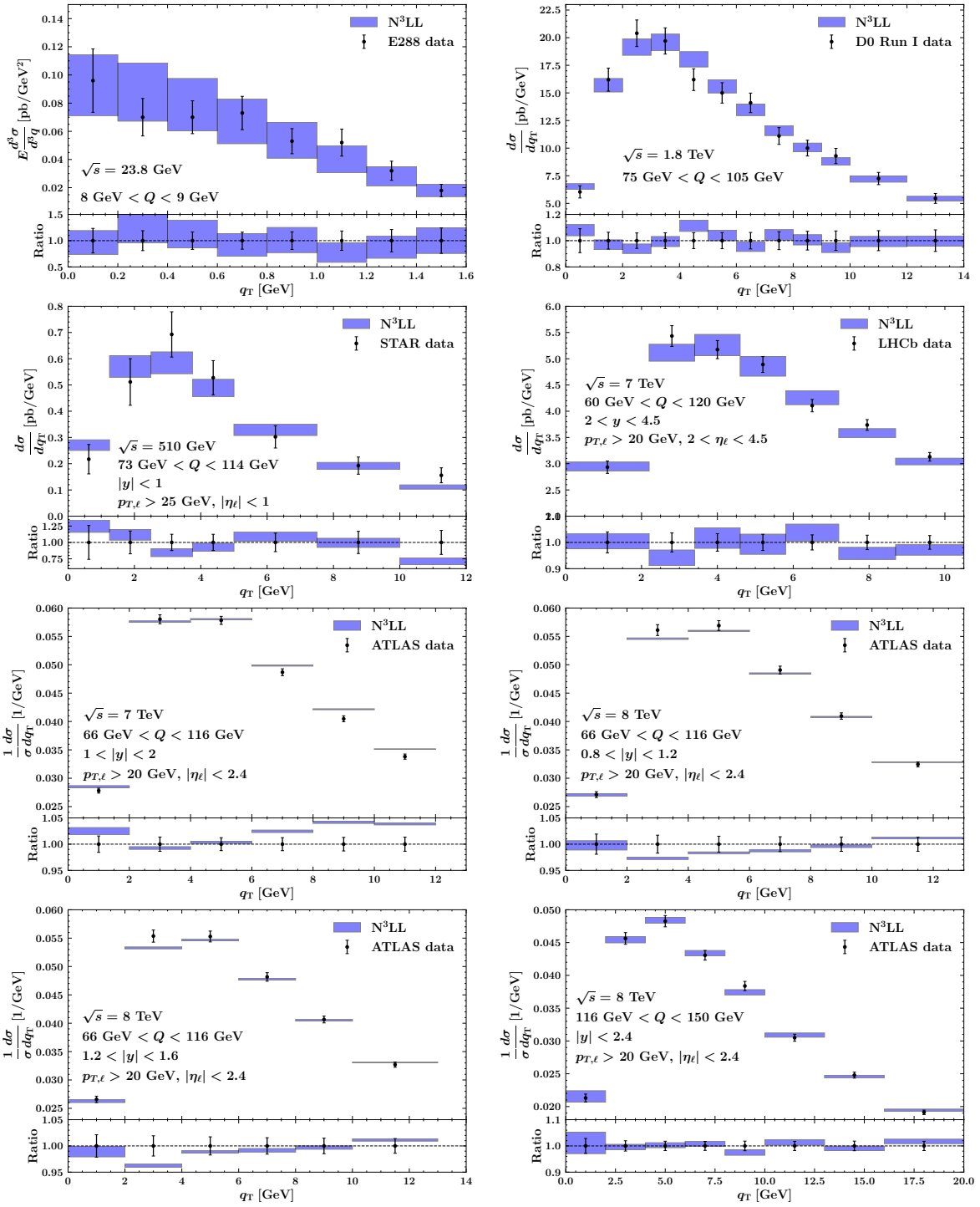


Figure 4.2: Comparison between experimental data and theoretical predictions obtained at N^{3}LL accuracy for a representative subset of the datasets included in this analysis. The upper panel of each plot displays the absolute q_T distributions, while the lower panel displays the same distributions normalized to the experimental central values. The blue bands represent the 1–σ uncertainty of the theoretical predictions.

4.3.1 Fit parameters

In order to assess the sensitivity of the experimental dataset to $f_{\rm NP}$, it is interesting to look at the values of the free parameters obtained from the fit. For the reader convenience, we rewrite here our choice of parameterization in the PV19 fit:

$$\begin{aligned} f_{\rm NP}(x, b_T, \zeta) &= \left[\frac{1-\lambda}{1+g_1(x)\frac{b_T^2}{4}} + \lambda \exp\left(-g_{1B}(x)\frac{b_T^2}{4}\right)\right] \\ &\times \exp\left[-\left(g_2 + g_{2B} b_T^2\right) \ln\left(\frac{\zeta}{Q_0^2}\right)\frac{b_T^2}{4}\right] , \end{aligned} \tag{4.4}$$

with $Q_0 = 1$ GeV and with the $g_1(x)$ and $g_{1B}(x)$ functions given by

$$\begin{aligned} g_1(x) &= \frac{N_1}{x\sigma} \exp\left[-\frac{1}{2\sigma^2} \ln^2\left(\frac{x}{\alpha}\right)\right] , \\ g_{1B}(x) &= \frac{N_{1B}}{x\sigma_B} \exp\left[-\frac{1}{2\sigma_B^2} \ln^2\left(\frac{x}{\alpha_B}\right)\right] . \end{aligned} \tag{4.5}$$

In Tab. 4.4 the average of each parameter over the Monte Carlo replicas, along with the respective standard deviation, is reported. All parameters are well constrained. We

Parameter	Value
g_2	0.036 ± 0.009
N_1	0.625 ± 0.282
α	0.205 ± 0.010
σ	0.370 ± 0.063
λ	0.580 ± 0.092
N_{1B}	0.044 ± 0.012
α_B	0.069 ± 0.009
σ_B	0.356 ± 0.075
g_{2B}	0.012 ± 0.003

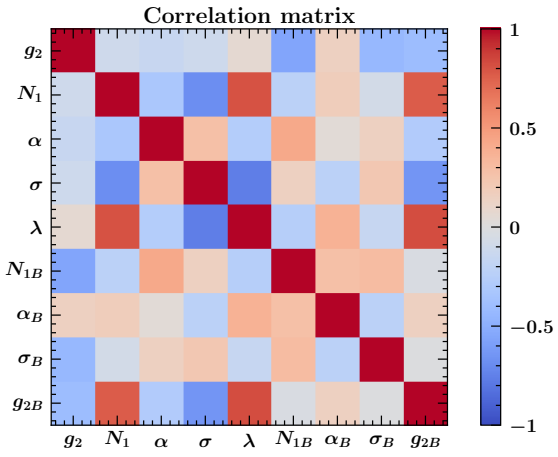


Table 4.4: Average and standard deviation over the Monte Carlo replicas of the free parameters fitted to the data and graphical representation of the correlation matrix.

stress that the parameters reported in Tab. 4.4 are not a direct result of any of our fits: they do not match any specific replica. It is interesting to observe that the parameter λ, that measures the relative weight of Gaussian and q-Gaussian in Eq. (2.42), is close to 0.5 indicating that these contributions weigh approximately the same. Concerning

the values of the parameters g_2 and g_{2B} associated to the nonperturbative contribution to TMD evolution, we find that the coefficient g_{2B} of the quartic term is small but significantly different from zero, as can be also seen in Fig. 4.3. This seems to suggest that higher–power corrections to the commonly assumed quadratic term g_2 may be required by the data.

A visual appraisal of the quality of the determination of the parameters are can be also done looking at Fig. 4.3, where we show how the parameter values of the PV19 200 replicas are distributed. In other words, Fig. 4.3 reflects the content of Tab. 4.4.

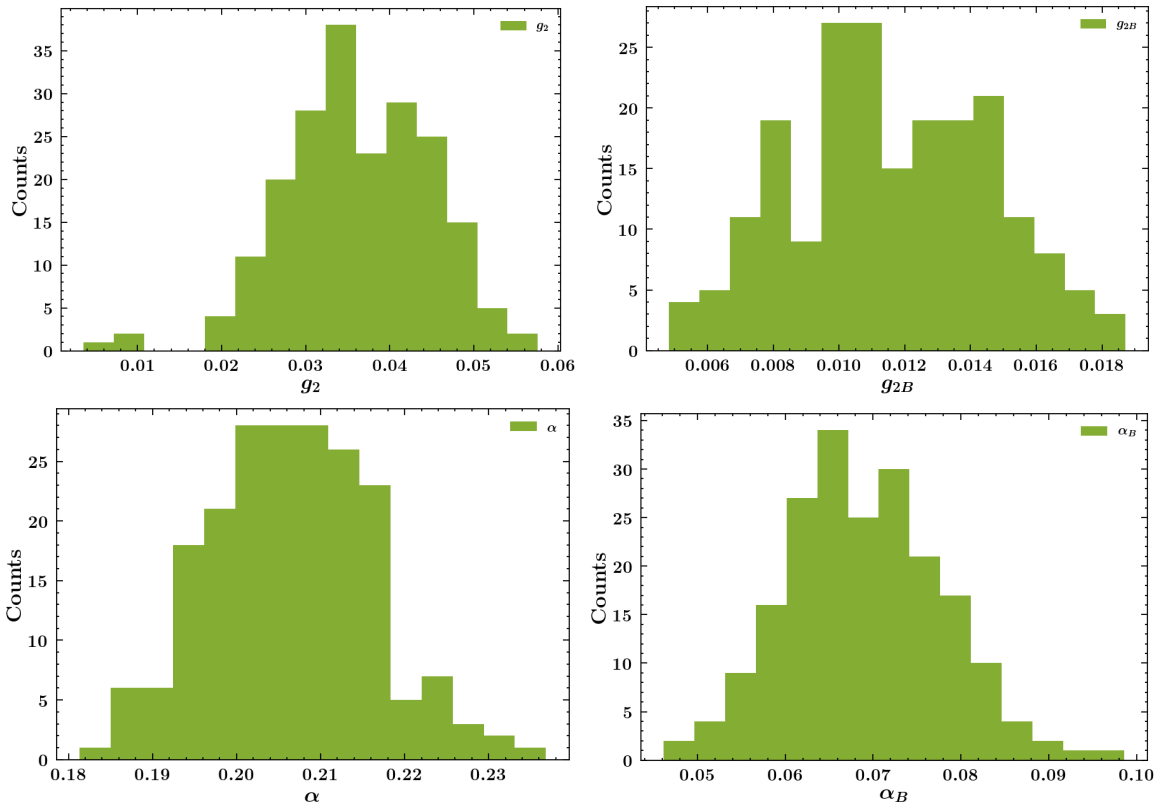


Figure 4.3: A selection of plots that represent the distributions of PV19 free parameters. The values of the 200 replicas plotted here are compatible with Gaussian distributions whose standard deviations are reported in Tab. 4.4. The situation is the same also for the other five free parameters.

Further insight concerning the appropriateness of the functional form in Eqs. (2.42)-(2.43) can be gathered by looking at the statistical correlations between parameters. In the right panel of Tab. 4.4, we show a graphical representation of the correlation matrix of the fitted parameters. The first observation is that (off–diagonal) correlations are generally not very large. There is however one exception: the parameters σ and λ seem to be strongly anti–correlated, and this may indicate that the interplay between

q-Gaussian and Gaussian may be significantly x dependent.

To conclude this section, we show the results of a scan in the parameter space performed with the `MINUIT` function `ROOT::Minuit2::MnScan`. The plots in Fig. 4.4 are created when running `fit.py` with the option "scan"and are an evidence of the fact that the parameter values we obtained are stable. The scan is performed in the following way: all parameters, except a selected one, are kept frozen at their final value; the selected parameter is then varied in an interval of ~ 2 standard deviations around its final value and the χ^2 is computed. In Fig. 4.4, we can see that the final parameters correspond to a minimum of the χ^2. In particular, we show that there are no flat directions in the parameter space, and therefore the minimum is particularly stable.

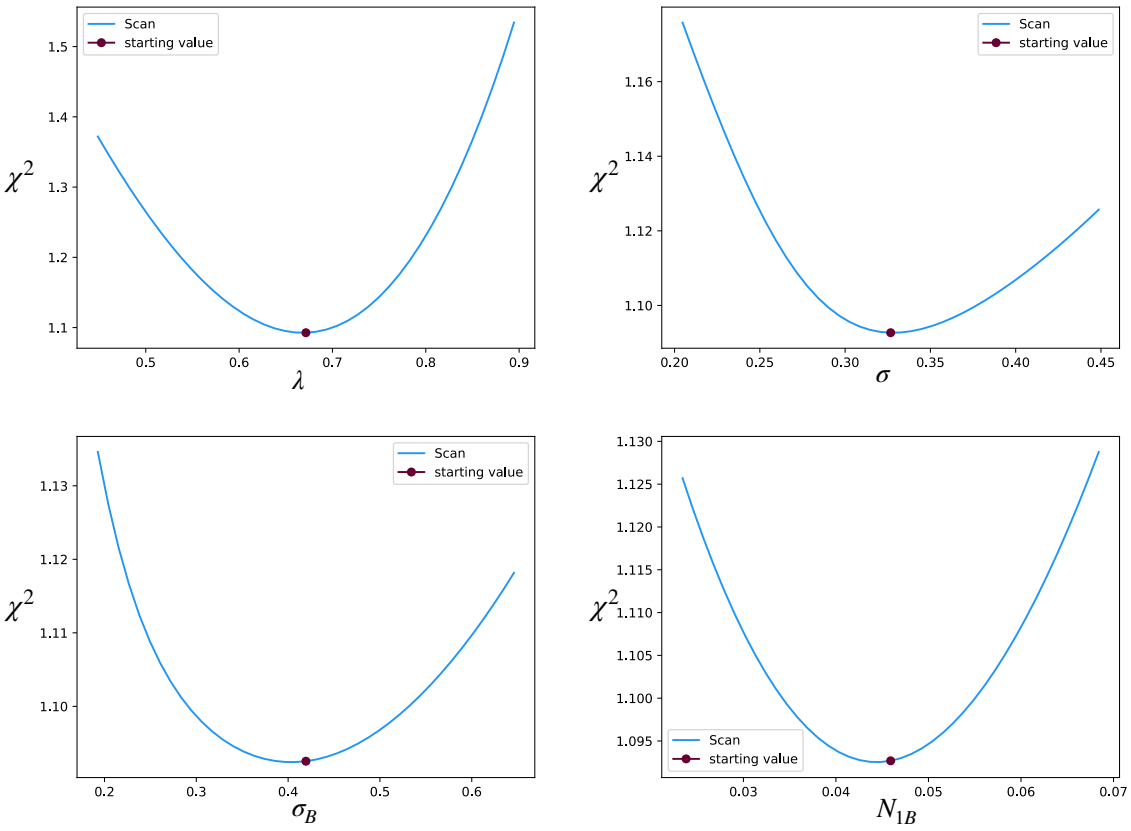


Figure 4.4: A selection of plots that represent how the χ^2 changes as $f_{\rm NP}$ parameters are varied one at a time. For this scan in the parameter space we used the values of the central replica (replica_0). "Starting value" refers to the value of the parameter before performing the scan, *i.e*, the final value of the parameter after the fit converged. The parameters $g_2, g_{2B}, \alpha, \alpha_B$ and N_1 present a situation of minimum similar to the one showed here, where it is clear that there are no flat directions in the parameter space and consequently the minimum appears stable.

4.3.2 TMD distributions

We discuss now the TMD distributions extracted from our reference N^{3}LL fit. We stress once again that only the combination in the right hand side of Eq. (2.39) is meaningful.

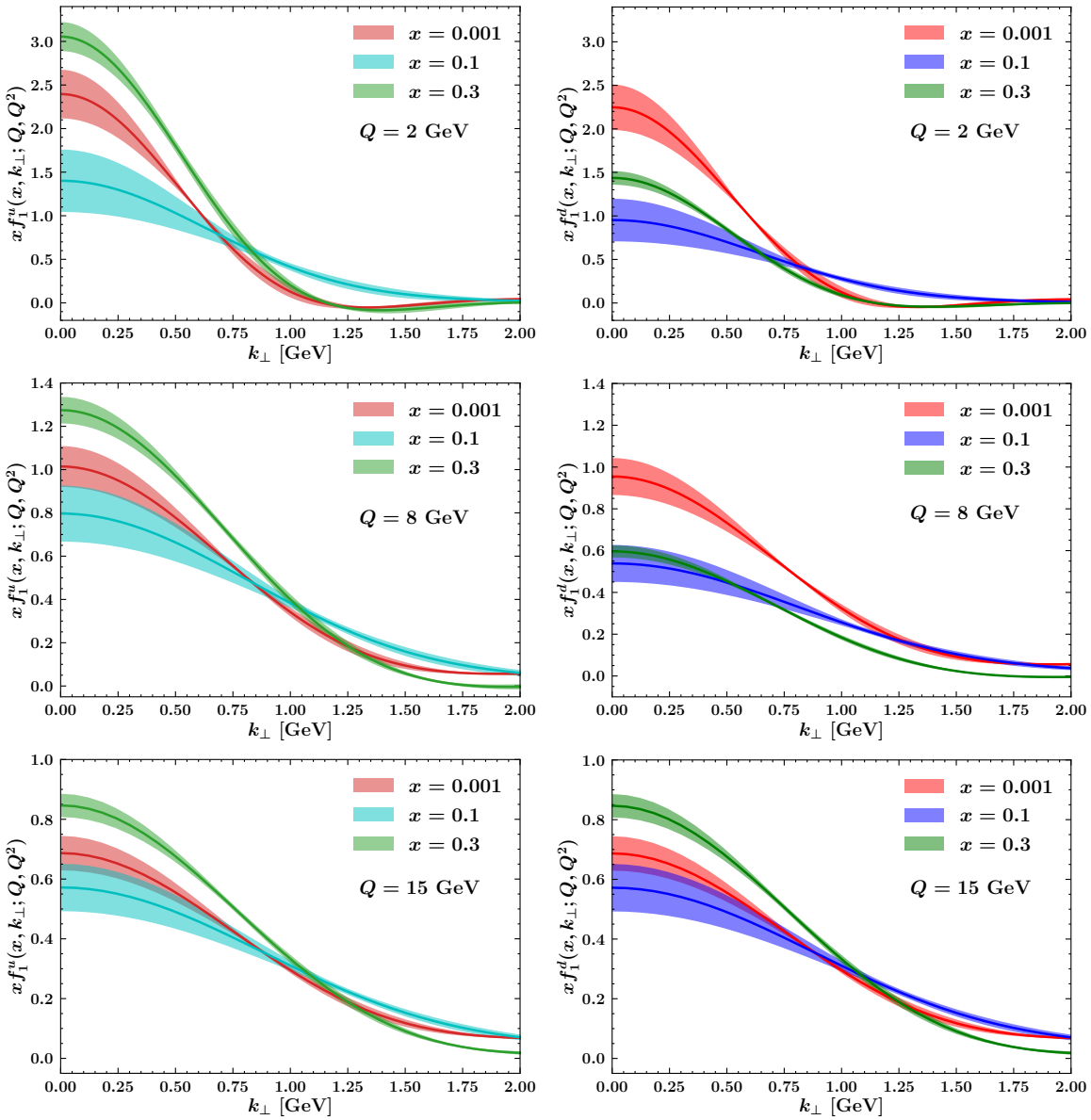


Figure 4.5: The TMD of the up and down quarks at different values of $\mu = \sqrt{\zeta} = Q$ [GeV] as a function of the partonic transverse momentum $k_\perp$ for different values of x. The bands give the 1-σ uncertainty.

In Fig. 4.5 we show the up and down quark TMDs at $Q = 2$ GeV, $Q = 8$ GeV and $Q = 15$ GeV as a function of the partonic transverse momentum $k_\perp$ for different values

of x. The 1-σ uncertainty bands are also shown. We show the same values of Q and x, allowing a direct comparison between the two. As can be observed comparing the plots in the first and in the last row of Fig. 4.5, the value at the peak of the curves decreases and the distributions become broader as Q increases. As expected, TMDs are suppressed as $k_\perp$ grows and the suppression becomes relatively stronger as Q increases.

4.4 Perturbative convergence

In the previous section we discussed the quality of our fit at N^{3}LL, which is the best accuracy presently available. In this section we show how the inclusion of perturbative corrections is crucial to achieve a better description of the experimental data. To this end, we performed fits at NLL′, NNLL, and NNLL′ (see Sec. 2.1.4), and compared them to the N^{3}LL fit. We did not consider LL and NLL accuracies because in both cases the description of the data is very poor ($\chi^2 \gtrsim 20$).

	NLL′	NNLL	NNLL′	N^{3}LL
Global χ^2	1126	571	379	360

Table 4.5: Values of the global χ^2 of the fits at NLL′, NNLL, NNLL′, and N^{3}LL accuracy.

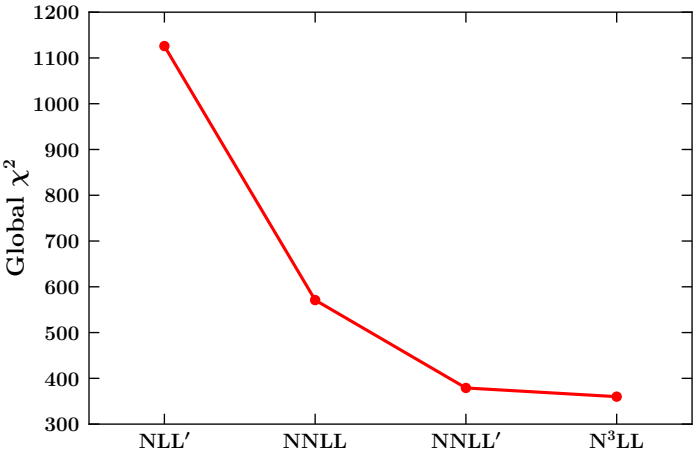


Figure 4.6: Graphical representation of Tab. 4.5.

Tab. 4.5 reports the values of the global χ^2 for each of the four accuracies considered. In order to appreciate the significance of the differences,[3] we have reported the

[3]Note that a difference of n units at the level of the global χ^2 roughly means a separation of around $\sqrt{n}$ standard deviations.

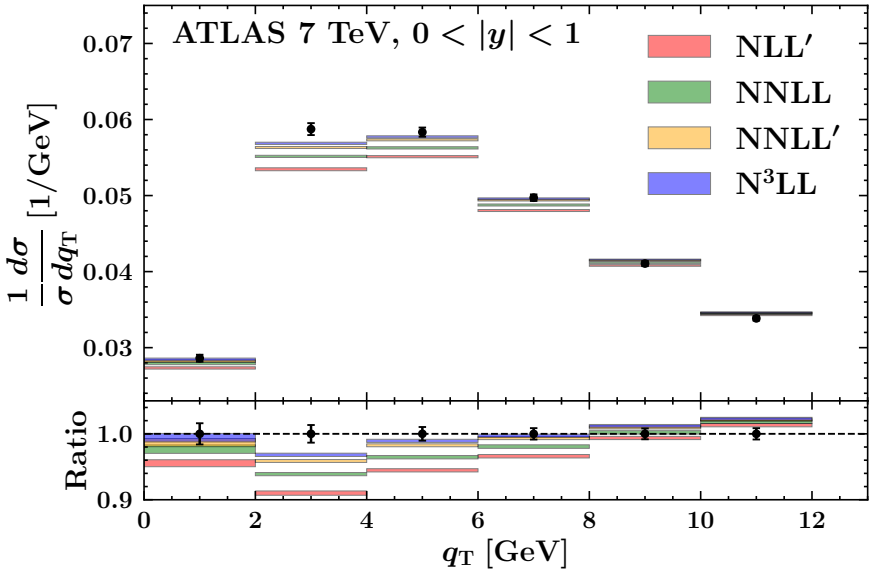


Figure 4.7: Comparison between experimental data for the ATLAS 7 TeV measurements in the bin 66 GeV $< Q <$ 116 GeV and $0 < |y| < 1$ and the theoretical predictions obtained from the fits to all perturbative orders considered in this analysis, *i.e.*, NLL′, NNLL, NNLL′, and N^{3}LL (see Sec. 2.1.4). The layout of the plot is the same as in Fig. 4.2.

absolute values of the χ^2 without dividing by the number of data points $N_{\rm dat}$. Fig. 4.6 shows a graphical representation of Tab. 4.5. The global quality of the fit improves significantly as the perturbative accuracy increases. In addition, Fig. 4.6 shows that the convergence rate decreases when going to larger perturbative orders. On the one hand, we conclude that it is necessary to include higher perturbative corrections to obtain a good description of the data and that N^{3}LL corrections are still significant. On the other hand, it appears that the perturbative series is nicely converging and N^{3}LL accuracy seems appropriate within the current experimental uncertainties.

In order to quantify the numerical impact of higher–order corrections, in Fig. 4.7 we compare the predictions for all the available perturbative orders to the ATLAS 7 TeV data in the bin 66 GeV $< Q <$ 116 GeV and $0 < |y| < 1$. This plot shows how the inclusion of higher–order corrections improves the shape of the predictions, particularly around the peak region.

4.5 Reduced dataset and x dependence

The nonperturbative function $f_{\rm NP}$, Eq. (2.39), accounts for the large–b_T behavior of TMDs and it is in general a function of b_T, ζ, and x. While the asymptotic dependence on b_T is driven by first–principle considerations (see Sec. 2.1.5) and the evolution

with ζ is determined by the Collins–Soper equation (2.11), the dependence on x is totally unknown. Moreover, a direct access to the x dependence is particularly difficult to achieve because it requires cross section data finely binned in rapidity y. In the dataset considered here, only the ATLAS experiment delivers data differential in rapidity. Therefore, one would expect that these datasets provide most of the sensitivity to the x dependence of TMDs.

In order to test this conjecture, we employed a particularly simple x–independent parameterization of the nonperturbative function:

$$f_{\rm NP}^{\rm DWS}(b_T,\zeta) = \exp\left[-\frac{1}{2}\left(g_1 + g_2 \ln\left(\frac{\zeta}{2Q_0^2}\right)\right) b_T^2\right] , \tag{4.6}$$

with two free parameters, g_1 and g_2, and $Q_0^2 = 1.6$ GeV^2 (inspired by the pioneering work of Davies, Webber, and Stirling. [67]). Using Eq. (4.6) we first performed a fit at N^{3}LL to the full dataset. Then we excluded the ATLAS datasets differential in rapidity (but we kept the off–peak ATLAS 8 TeV datasets because inclusive in rapidity). The

	Full dataset	No y-differential data
Global $\chi^2/N_{\rm dat}$	1.339	0.895
g_1	0.304	0.207
g_2	0.028	0.093

Table 4.6: The values of the global χ^2 normalized to the number of data points $N_{\rm dat}$ from the fit to the full dataset and to a reduced dataset without the y–differential ATLAS datasets, both using the parameterization in Eq. (4.6). For completeness, we also report the best–fit values of the parameters g_1 and g_2.

resulting χ^2s normalized to the number of data points are reported in Tab. 4.6. For completeness, we also show the best–fit values of the parameters g_1 and g_2.

Firstly, the χ^2 of the fit to the full dataset using Eq. (4.6) is equal to 1.339 and is significantly larger than that obtained using the parameterization in Eqs. (2.42)-(2.43) (1.020). This suggests that an x–dependent $f_{\rm NP}$ is required to obtain a good description of the data. Secondly, the χ^2 of the fit without the y–differential ATLAS data comes out to be particularly low (0.895). We conclude that at N^{3}LL accuracy the x dependence of the TMDs extracted from the currently available DY data is mostly required by the ATLAS data differential in the boson rapidity y. We note however that the agreement with the very precise ATLAS data may be influenced also by other small corrections (*e.g.* power corrections).

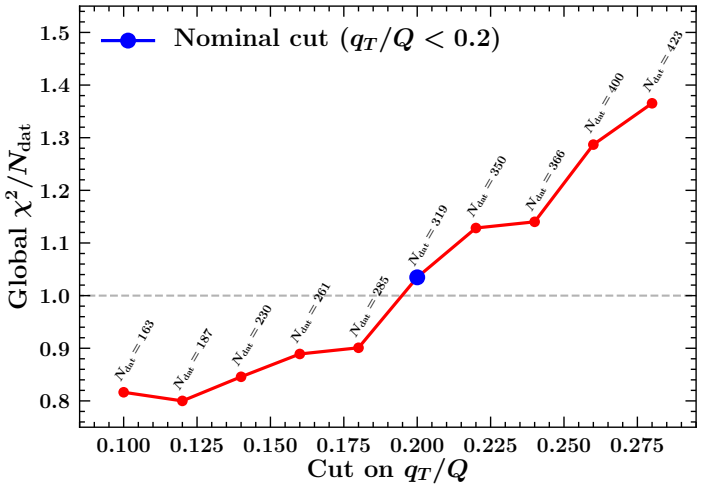


Figure 4.8: The global $\chi^2/N_{\rm dat}$ as a function of the cut on q_T/Q. The blue point corresponds to the reference cut used in this analysis.

4.6 Dependence on the cut on q_T/Q

As discussed in Ch. 2, our analysis is based on TMD factorization whose validity is restricted to the region $q_T \ll Q$. As a consequence, we consider only measurements that respect this constraint. More precisely, we require that the maximum value of the ratio q_T/Q for a point to be included in the fit be 0.2 (see Sec. 3.3). Despite this particular value seems to be generally recognized in the literature (see, *e.g.*, Ref. [78]), it is interesting to study how the global description of the dataset changes by varying this cut. This can also help us assess more quantitatively the validity range of TMD factorization.

Fig. 4.8 displays the behavior of the global $\chi^2/N_{\rm data}$ for the N^{3}LL fit as a function of the q_T/Q cut ranging between 0.1 and 0.28 in steps of 0.02. As expected, the quality of the fit tends to degrade as the cut on q_T/Q increases. Of course, it is impossible to draw a line between validity and non–validity regions. However, this study gives a quantitative justification for choosing the value 0.2 for the q_T/Q cut.

4.7 PV19 fit results

In Fig. 4.9 we show the χ^2 values of the 200 Monte Carlo replicas of the PV19 TMD PDF extraction. As expected, the shape of the histogram is a χ^2 distribution. The fact that such distribution has its peak in a value very close to 1 is an indication of optimal fit quality.

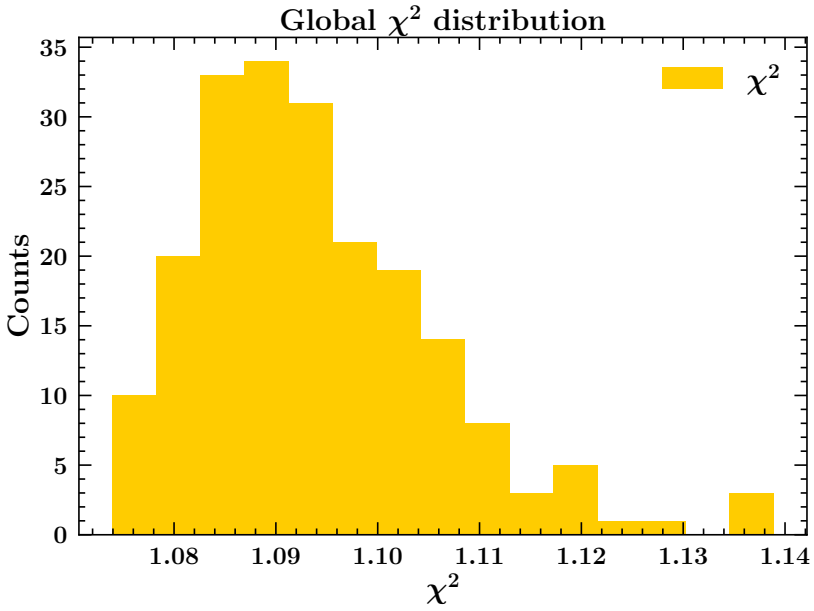


Figure 4.9: Distribution of the χ^2 values of the 200 Monte Carlo replicas of the PV19 TMD extraction.

In Tab. 4.7 we show the χ^2s computed as an average over the 200 replicas we considered in our fit and its standard deviation. In this case, separating uncorrelated and correlated contributions to the total χ^2 is not useful, as $\langle\chi^2_\lambda\rangle + \langle\chi^2_D\rangle \neq \langle\chi^2_{\text{tot}}\rangle$.

The value of the total χ^2 is, as expected, slightly different from the χ^2 computed with respect to the mean of the ensemble of replicas (in Tab. 4.3): the fact that also the average χ^2 is very close to 1 (1.07), suggests a good stability of our results and is another indicator of the goodness of our TMD PDF extraction. Note that the global χ^2 distribution in Fig. 4.9 is peaked around 1.07, the value of $\langle\chi^2_\lambda\rangle$ reported in Tab. 4.3.

As for the χ^2 values of each experiment, the same considerations done in Sec. 4.3 for the case of χ^2 computed with respect to to the mean replica hold also for Tab. 4.7. In general, LHC data, and in particular ATLAS data, have higher values of χ^2, but, given the extremely high precision of these datasets, the χ^2 values reported in Tab. 4.7 are to be considered a very good result.

Finally, we report a selection of plots (Fig. 4.10–Fig. 4.21) that show all the Monte Carlo replicas and the distribution of their χ^2 for single experiments. In the left panels, the χ^2 are calculated with respect to the mean replica, which is highlighted in red in the right panels: the histograms approximate χ^2 distributions peaked around the values reported in Tab. 4.3. In the right panels, the black line refers to the central replica, *i.e*, replica 0. Note that, for low–energy experiments, the width of the replica bundle is bigger than replicas are more spread for high–energy experiments (see LHC plots, for

Experiment	Q[GeV], y	χ^2
E605	7 GeV $< Q <$ 8 GeV	0.69 ± 0.11
	8 GeV $< Q <$ 9 GeV	1.54 ± 0.22
	10.5 GeV $< Q <$ 11.5 GeV	0.55 ± 0.07
	11.5 GeV $< Q <$ 13.5 GeV	1.39 ± 0.11
	13.5 GeV $< Q <$ 18 GeV	2.04 ± 0.11
E288 (200)	4 GeV $< Q <$ 5 GeV	1.43 ± 0.29
	5 GeV $< Q <$ 6 GeV	1.67 ± 0.20
	6 GeV $< Q <$ 7 GeV	0.42 ± 0.08
	7 GeV $< Q <$ 8 GeV	0.43 ± 0.14
	8 GeV $< Q <$ 9 GeV	0.60 ± 0.07
E288 (300)	4 GeV $< Q <$ 5 GeV	1.09 ± 0.27
	5 GeV $< Q <$ 6 GeV	0.94 ± 0.14
	6 GeV $< Q <$ 7 GeV	0.46 ± 0.13
	7 GeV $< Q <$ 8 GeV	0.12 ± 0.04
	8 GeV $< Q <$ 9 GeV	0.58 ± 0.10
	11 GeV $< Q <$ 12 GeV	0.61 ± 0.03
E288 (400)	5 GeV $< Q <$ 6 GeV	0.32 ± 0.08
	6 GeV $< Q <$ 7 GeV	0.11 ± 0.03
	7 GeV $< Q <$ 8 GeV	0.08 ± 0.05
	8 GeV $< Q <$ 9 GeV	0.50 ± 0.08
	11 GeV $< Q <$ 12 GeV	0.60 ± 0.14
	12 GeV $< Q <$ 13 GeV	0.57 ± 0.04
	13 GeV $< Q <$ 14 GeV	0.70 ± 0.05
STAR		0.93 ± 0.05
CDF RunI		0.59 ± 0.02
CDF RunII		0.90 ± 0.07
D0 RunI		0.67 ± 0.03
D0 RunII		1.70 ± 0.22
D0 RunII(μ)		3.21 ± 0.44
LHCb (7 TeV)		1.28 ± 0.04
LHCb (8 TeV)		0.67 ± 0.13
LHCb (13 TeV)		0.85 ± 0.07
CMS (7 TeV)		2.13 ± 0.01
CMS (8 TeV)		1.43 ± 0.05
ATLAS (7 TeV)	$0 < \lvert y \rvert < 1$	2.71 ± 0.24
	$1 < \lvert y \rvert < 2$	5.29 ± 0.09
	$2 < \lvert y \rvert < 2.4$	3.92 ± 0.11
ATLAS (8 TeV))	$0 < \lvert y \rvert < 0.4$	2.36 ± 0.13
	$0.4 < \lvert y \rvert < 0.8$	2.50 ± 0.09
	$0.8 < \lvert y \rvert < 1.2$	0.99 ± 0.08
	$1.2 < \lvert y \rvert < 1.6$	1.04 ± 0.15
	$1.6 < \lvert y \rvert < 2$	0.80 ± 0.24
	$2 < \lvert y \rvert < 2.4$	1.05 ± 0.22
ATLAS (8 TeV)	46 GeV $< Q <$ 66 GeV	2.99 ± 0.12
	116 GeV $< Q <$ 150 GeV	0.50 ± 0.01
Total		1.07

Table 4.7: Values of χ^2 computed as an average over the 200 replicas we considered in our fit.

example): this is due to the fact that E288 and E605 are affected by large normalization errors.

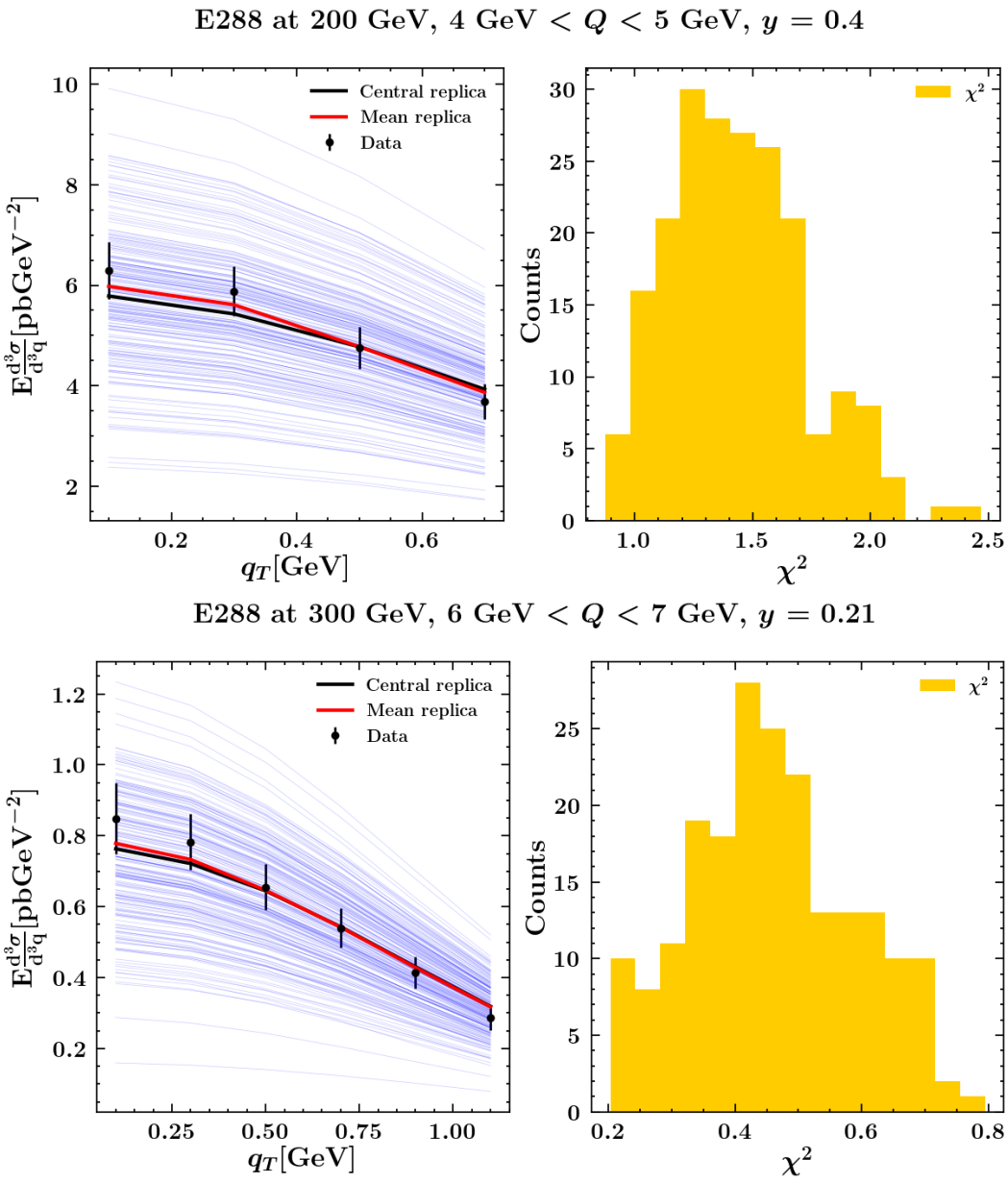


Figure 4.10: The left panels show the data points with their error bars and the 200 replicas of the PV19 fit for one selected dataset of E288 at 200 GeV and of E288 at 300 GeV: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the selected datasets.

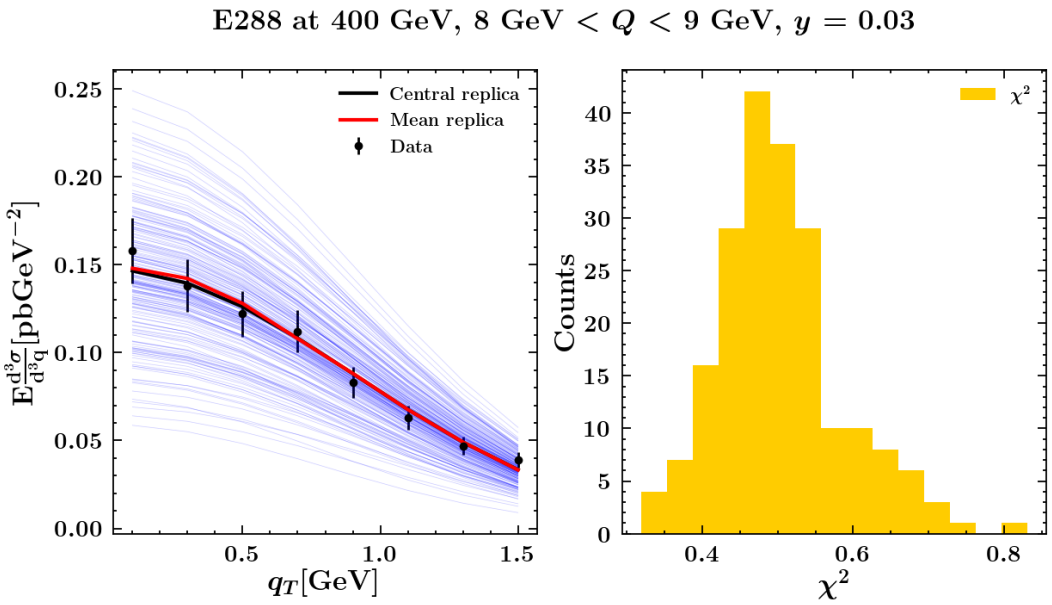


Figure 4.11: The left panel shows the data points with their error bars and the 200 replicas of the PV19 fit for one selected dataset of E288 at 400 GeV: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panel we display the χ^2 distribution of all the replicas for the selected dataset.

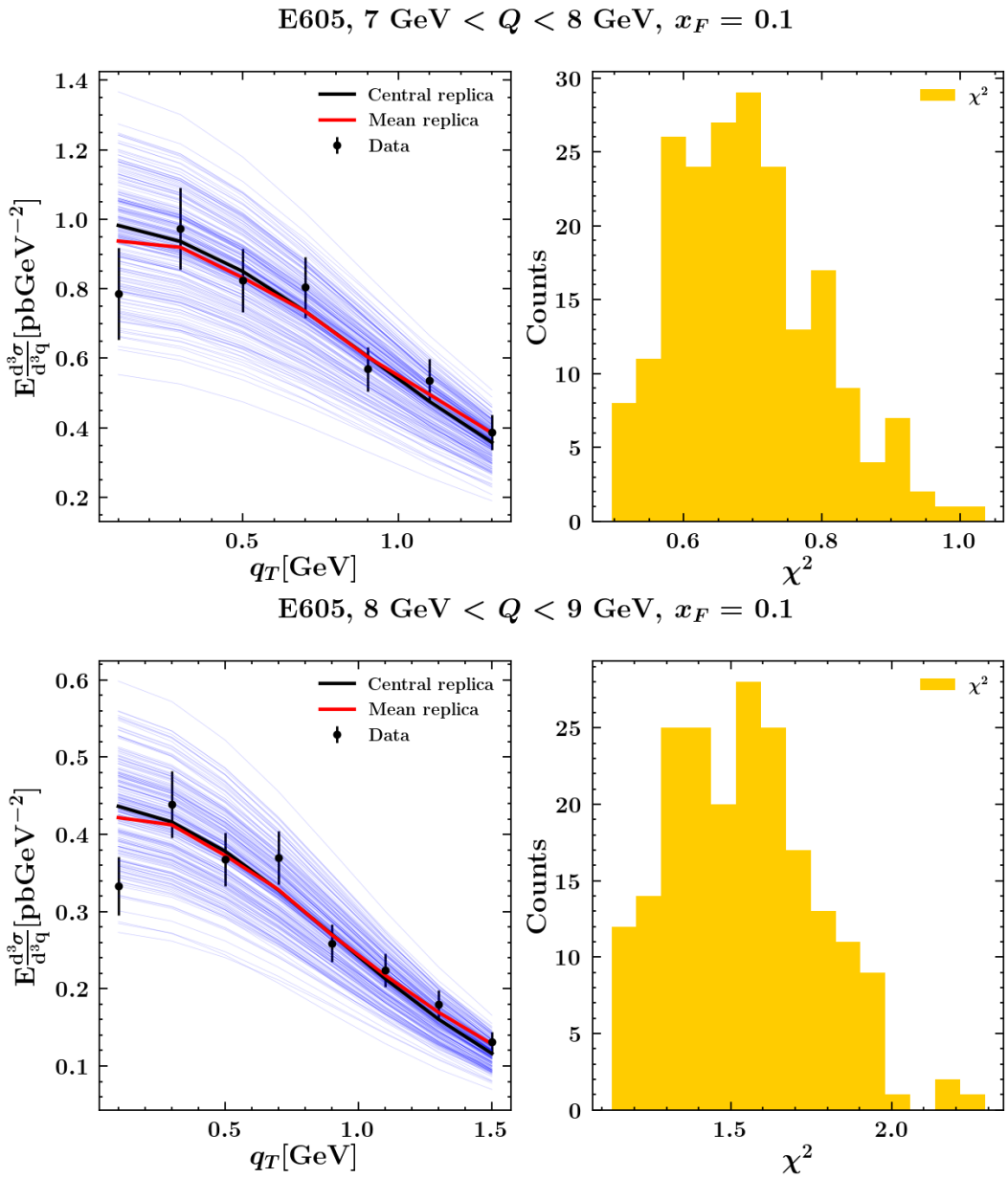


Figure 4.12: The left panels show the data points with their error bars and the 200 replicas of the PV19 fit for two selected dataset of E605 at different values of Q: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the selected datasets.

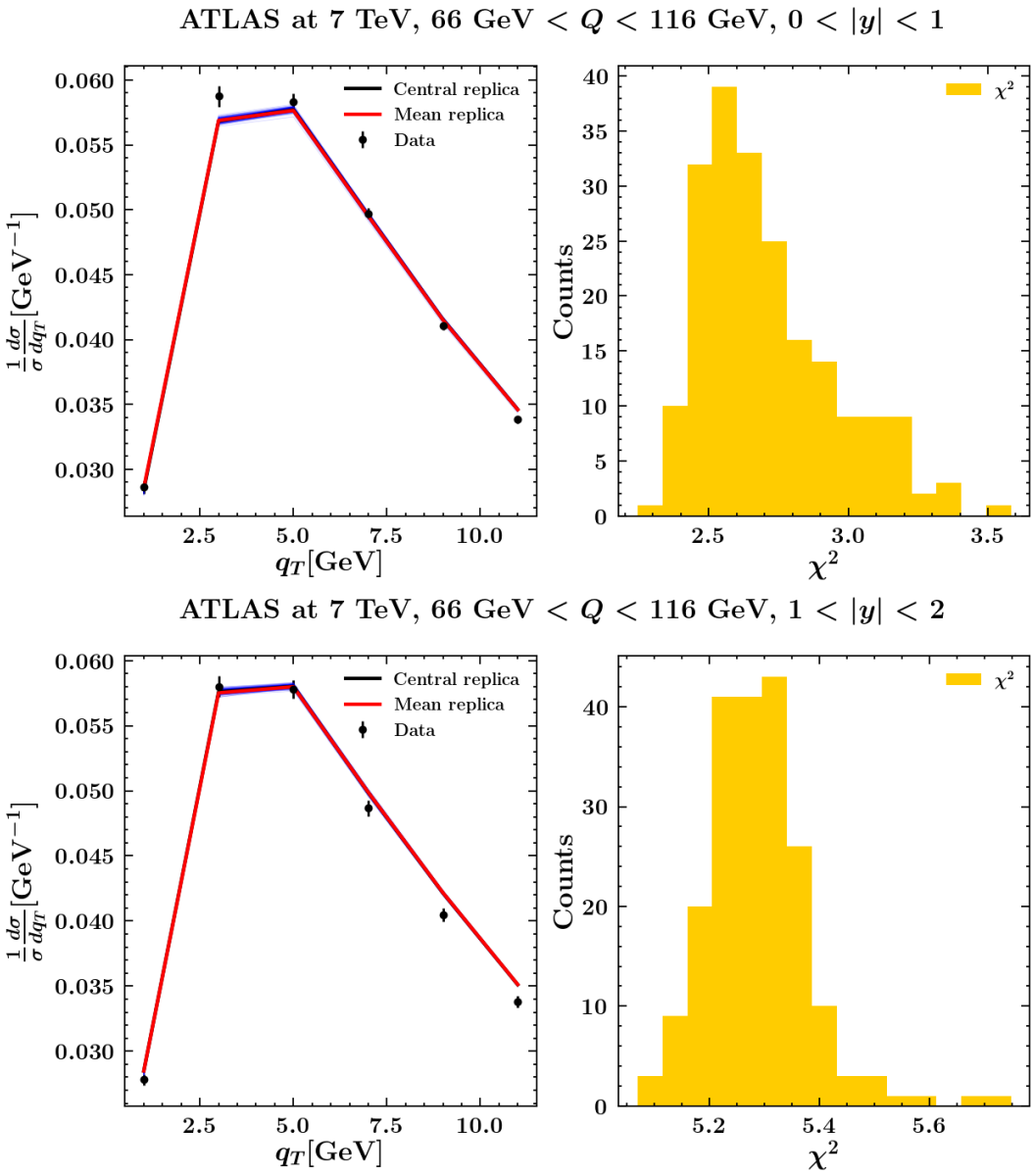


Figure 4.13: The left panels show the data points with their error bars and the 200 replicas of the PV19 fit for two y–bins of ATLAS at 7 TeV: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the datasets.

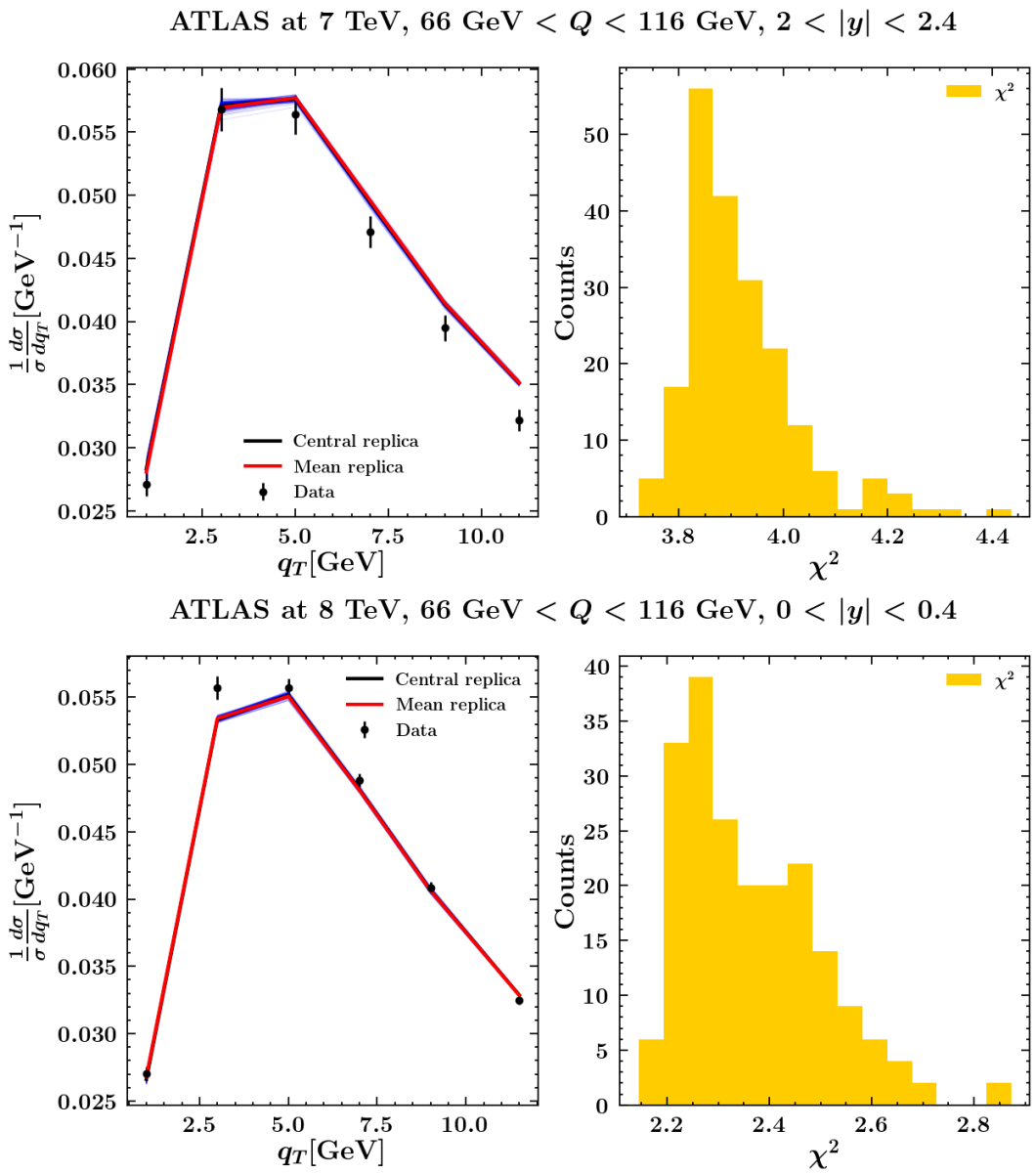


Figure 4.14: The left panels show the data points with their error bars and the 200 replicas of the PV19 fit for the highest y–bin available for ATLAS at 7 TeV and the central y–bin of ATLAS at 8 TeV: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the datasets.

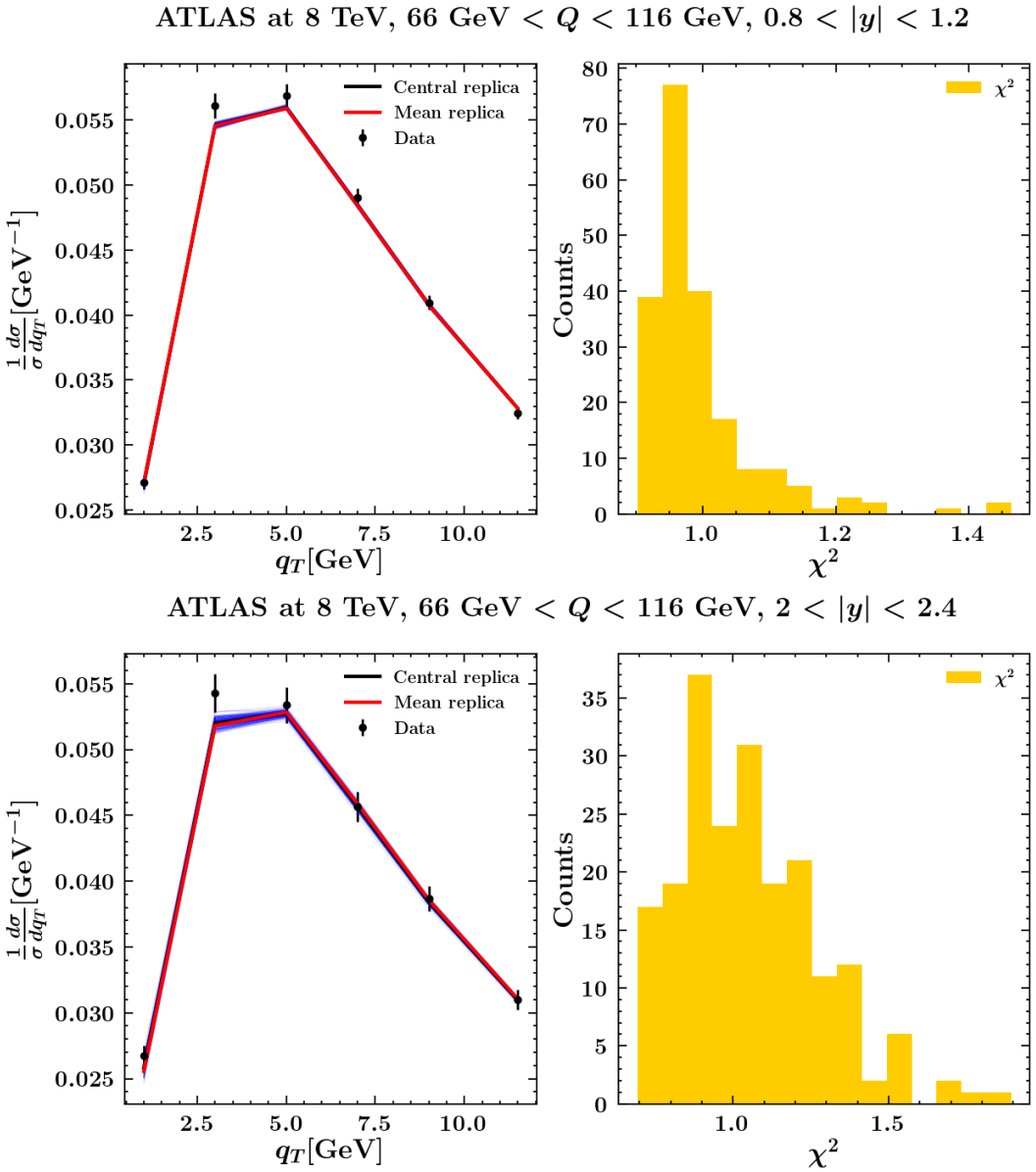


Figure 4.15: The left panels show a comparison between the data points with their error bars and the 200 replicas of the PV19 fit for two of the y–bins available for ATLAS at 8 TeV: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the datasets.

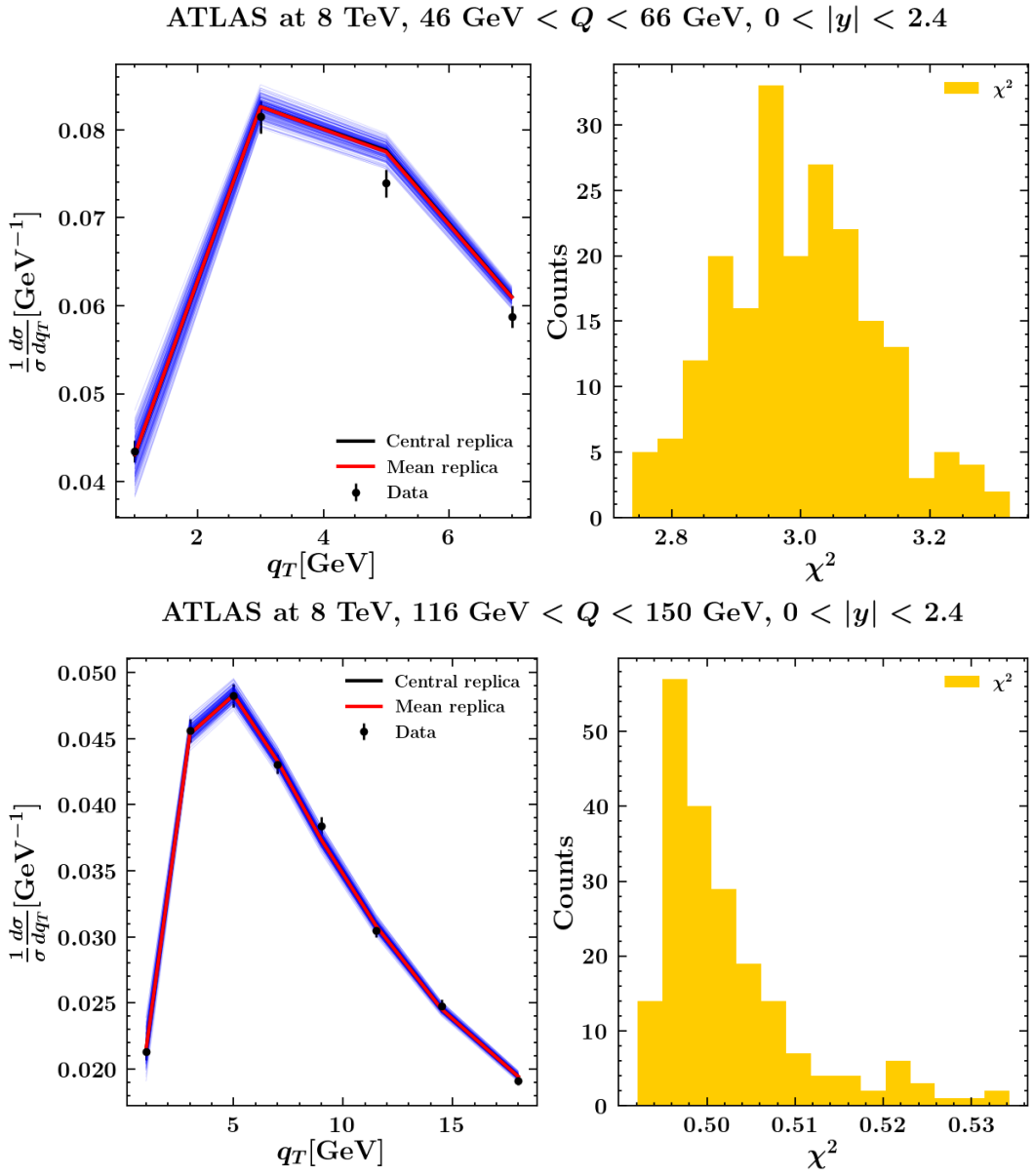


Figure 4.16: The left panels show a comparison between the data points with their error bars and the 200 replicas of the PV19 fit for ATLAS at 8 TeV off–peak: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the datasets.

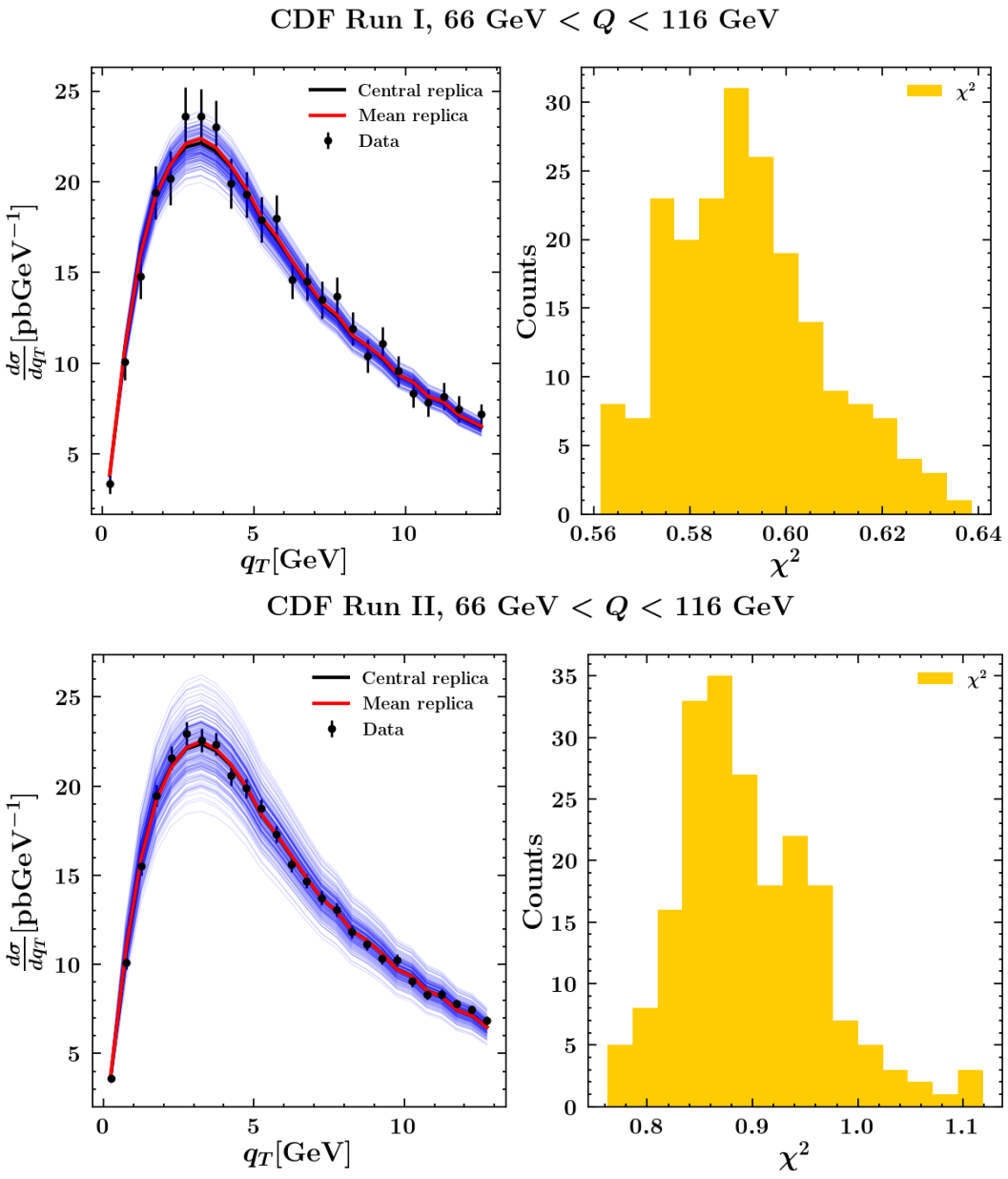


Figure 4.17: The left panels show a comparison between the data points with their error bars and the 200 replicas of the PV19 fit for CDF Run I and CDF Run II: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the datasets.

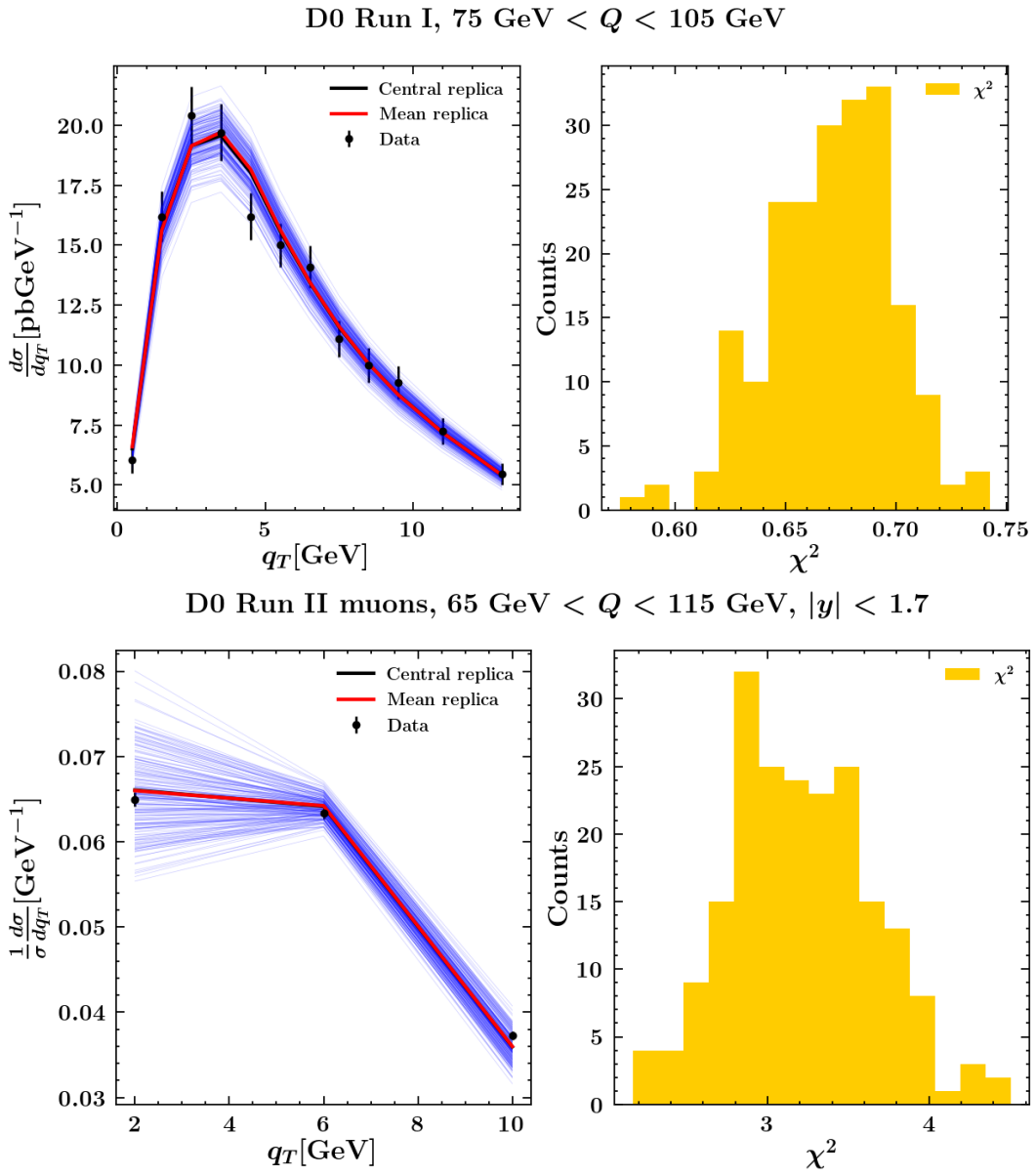


Figure 4.18: The left panels show a comparison between the data points with their error bars and the 200 replicas of the PV19 fit for D0 Run I and D0 Run II with muons: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the datasets.

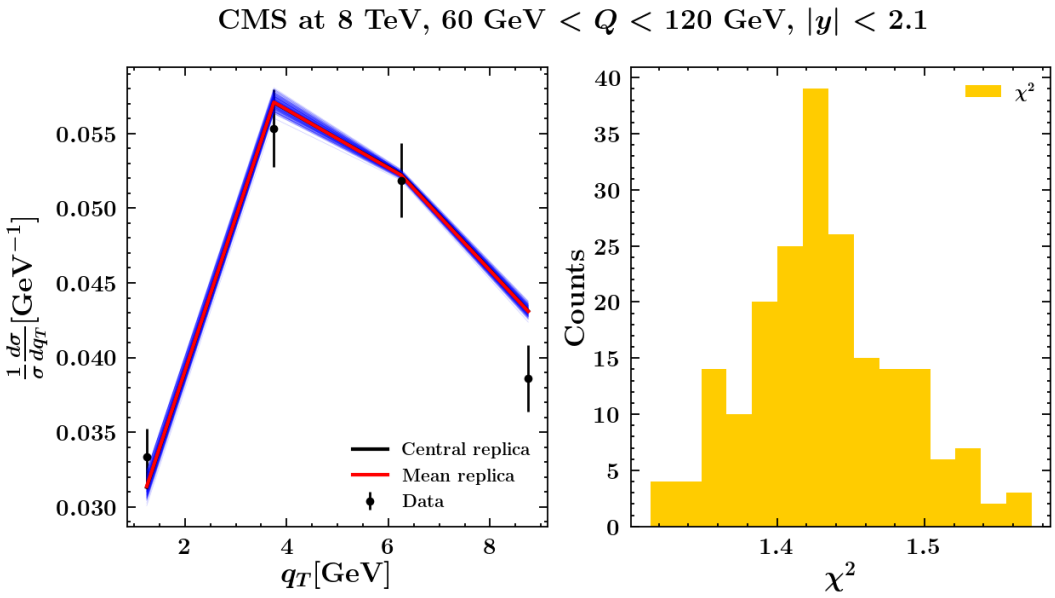


Figure 4.19: The left panel shows a comparison between CMS at 8 TeV data points and the 200 replicas of the PV19 fit: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panel we display the χ^2 distribution of all the replicas for the dataset.

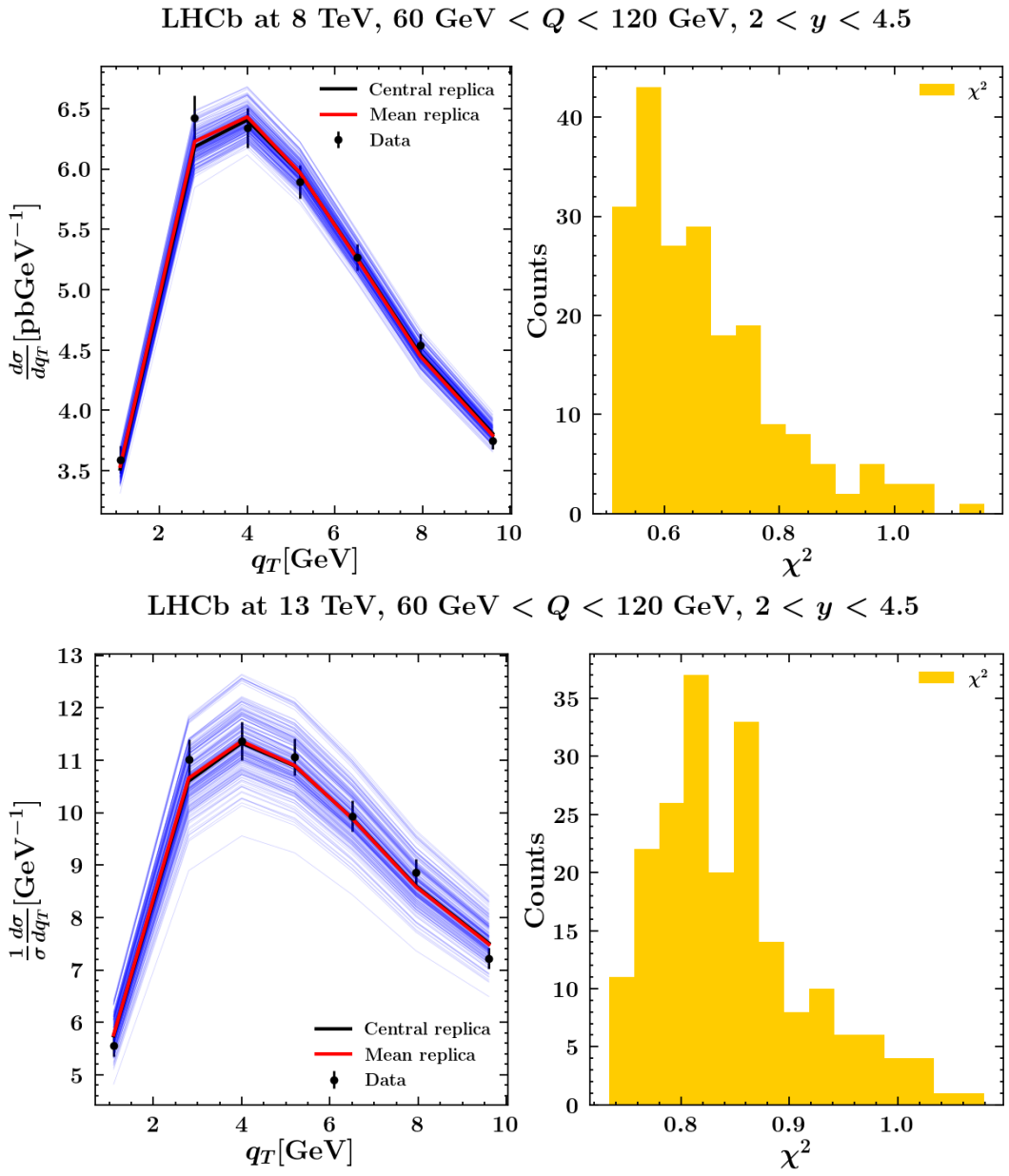


Figure 4.20: The left panels show a comparison between the data points with their error bars and the 200 replicas of the PV19 fit for LHCb at 8 TeV and 13 TeV: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panels we display the χ^2 distribution of all the replicas for the datasets.

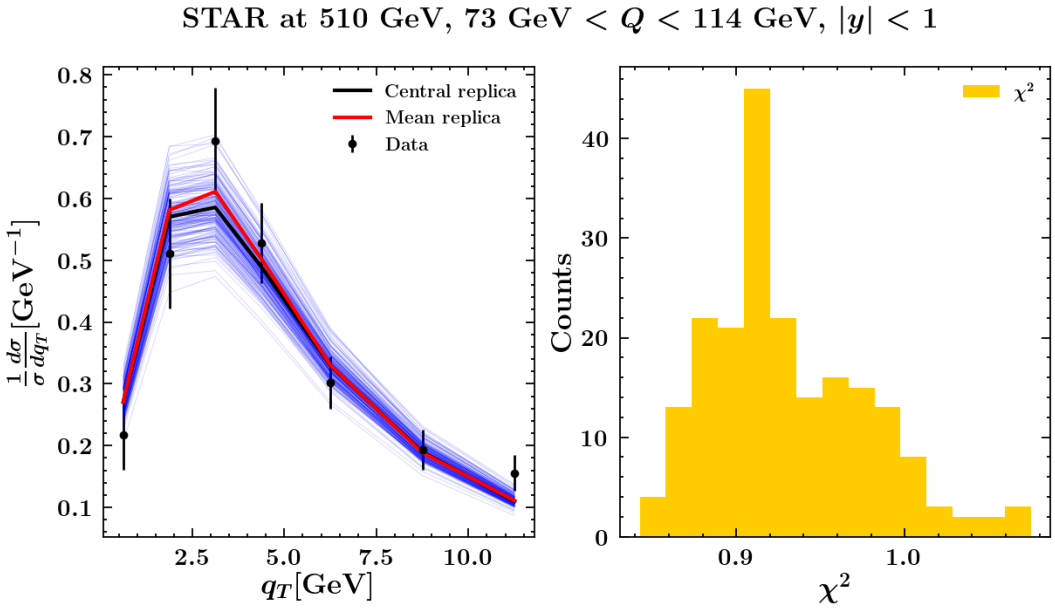


Figure 4.21: The left panel shows a comparison between the STAR data points we considered and the 200 replicas of the PV19 fit: the red line refers to the mean replica, while the black line is the central replica, *i.e* the replica that fits the original, non–fluctuated data points. In the right panel we display the χ^2 distribution of all the replicas for the dataset.

Chapter 5

Conclusions

The main topics of this thesis were the description of the suite of computational tools that we developed for TMD studies (`NangaParbat`) and the extraction of TMD PDFs performed using NangaParbat and dubbed PV19 [95]. The author of this thesis was one of the main developers of the `NangaParbat` code and one of the main authors of the PV19 TMD extraction.

In this thesis, we studied the description of two processes: Drell–Yan, which is lepton pair production through virtual photon or Z–boson exchange, $h_1(P_1) + h_2(P_2) \longrightarrow \gamma^*/Z(q) + X \longrightarrow \ell^+(l) + \ell^-(l') + X$, and Semi–Inclusive Deep Inelastic Scattering (SIDIS), an inelastic scattering of an high–energy lepton on an hadron target where one hadron in the final state is detected, $\ell(l) + N(P) \to \ell(l') + h(P_h) + X$.

Factorization theorems allow us to write the cross section of these processes in terms of Transverse Momentum Distributions (TMDs), as discussed in Ch. 2, where both perturbative and nonperturbative aspects of TMDs have been examined. In particular, we illustrated TMD evolution equations, discussing our choices for the values of the scales ($\mu = Q$ and $\zeta_1 = \zeta_2 = Q^2$), and we examined the perturbative ingredients that result from this particular choice. We explained how the resummation of large logarithms of Q/q_T is treated and how we defined the logarithmic ordering that quantifies the perturbative accuracy of a TMD extraction.

Then we considered the nonperturbative part of TMDs $f_{\rm NP}$ and its parameterization, introducing the b_*–prescription to regularize the behavior of the TMDs at large b_T and discussing two different nonperturbative functional forms: the one we used in our TMD Parton Distribution Function (TMD PDF) extraction in Ref. [95] and the choices made for the TMD PDF and TMD Fragmentation Function (TMD FF) extraction in Ref. [84] (PV17).

To perform the PV19 TMD PDF fit, we developed a C++ suite of tools for TMD physics, which we called `NangaParbat` and is publicly available at the GitHub reposi-

tory:

`https://github.com/MapCollaboration/NangaParbat`

The structure of `NangaParbat` is illustrated in Ch. 3. The purpose of `NangaParbat` is two–folded: it can be used to perform TMD extractions and to produce and interpolate TMD and structure function grids.

We illustrated how the TMD theoretical framework necessary to perform a TMD extraction is implemented in `NangaParbat`. In order to speed up the fit we precomputed and stored all perturbative (and collinear nonperturbative) terms. Thanks to this method, we reduced by a factor of ~ 200 the computation time required for the fit. Moreover, we discussed the processing of experimental data, the treatment of the uncertainties and the computation of the χ^2, which are crucial features of `NangaParbat` and were fundamental in obtaining a very good agreement with experimental data.

`NangaParbat` uses `APFEL++` as engine for calculations and makes use of advanced and optimized computation tools, which allowed us to perform *exactly* the multidimensional integrals necessary to produce predictions comparable with experimental data.

In Ch. 3 we also illustrated the `NangaParbat` tools that allow us to produce, interpolate and convolute TMD grids. In particular, `NangaParbat` is able to produce grids for unpolarized TMD PDFs, TMD FFs and for the unpolarized structure function $F_{UU,T}$. At the moment of the writing of this thesis, the following grids, which the author of this thesis significantly contributed to produce, are publicly available: PV19 TMD PDFs, PV17 TMD PDFs, PV17 TMD FFs for π^+ and π^- and PV17 structure function $F_{UU,T}(x, z, q_T, Q)$ for π^+ and π^- as hadrons detected in the final state. All these grids have been thoroughly tested and, together with the `NangaParbat` interpolator, have been included in TMDlib [193, 194], a library for TMD extractions. The grids can be found at

`https://syncandshare.desy.de/index.php/s/GjjcwKQC93M979e?path=%2FTMD%20grid%20files`

TMDlib has a plotting interface, TMDplotter, which can be used at

`http://tmdplotter.desy.de/TMDplotter.php?mode=default&type=tmdkt`.

In Ch. 4 we presented an extraction of TMDs from Drell-Yan data accurate up to N^{3}LL. We considered unpolarized observables that are accessible in a relatively large number of experiments and that allow us to extract the quark unpolarized TMD distributions and the NP part of TMD evolution. The dataset used in this analysis includes low–energy data from FNAL (E605 and E288) and RHIC (STAR) and high–energy data from Tevatron (CDF and D0) and the LHC (LHCb, CMS, and ATLAS), for a total of 353 data points.

The fit was performed with a proper treatment of the experimental uncertainties, which were propagated into the fitted TMD distributions by means of the Monte Carlo sampling method. This allowed us to obtain a very good description of the entire dataset ($\chi^2/N_{\rm dat} = 1.02$) without the need of introducing *ad hoc* normalizations. A more detailed analysis of the fit quality shows that both low– and high–energy datasets are separately well described. This is a remarkable achievement given the very high precision of the LHC datasets, especially those from ATLAS.

A particularly interesting aspect of our analysis concerns the QCD convergence of the perturbative series. We performed fits at NLL′, NNLL, NNLL′, and N^{3}LL accuracy and showed that the fit quality improves significantly going from NLL′ to N^{3}LL. The difference between the highest orders, *i.e.*, NNLL′ and N^{3}LL, is moderate but still significant. This shows at the same time that the perturbative series is converging, but also that N^{3}LL corrections are relevant in relation to the current experimental uncertainties.

We parameterized the nonperturbative contributions by adopting a reasonably flexible functional form: all nine free parameters turned out to be well constrained, with moderate correlations among them. An important feature of our parameterization of the nonperturbative contribution $f_{\rm NP}$ is its explicit x dependence. We proved that the x–dependent part of $f_{\rm NP}$ is mostly constrained by the rapidity–dependent on–peak data at 7 and 8 TeV from ATLAS. While on the one hand, this was to be expected because the x dependence is strictly connected with the rapidity y, on the other hand it also demonstrates that most of the datasets are not sensitive to the x dependence of TMDs.

Finally, we studied the validity range of TMD factorization in Drell–Yan by varying the cut on q_T/Q. In line with the literature, we found that the region $q_T \lesssim 0.2\,Q$ is appropriate when working within the TMD factorization framework.

5.1 Outlook

In this thesis we set the foundation for a number of future studies. In the first place, we plan to extend the fitted dataset by including the abundant and precise Semi-Inclusive DIS data from HERMES [139] and COMPASS [140, 166], as well as future data from Jefferson Lab at 12 GeV [218]. In Fig. 5.1 we show the kinematic coverage in the x–Q^2 plane of the Drell–Yan data we analyzed in this thesis and some of the present and future datasets from Semi–Inclusive DIS experiments. The Semi–Inclusive DIS data from the HERMES and COMPASS collaborations shown in Fig. 5.1 are already available and have been included in unpolarized TMD extractions in the literature [78, 84, 85]. As for future datasets, we report the expected kinematic range of the data

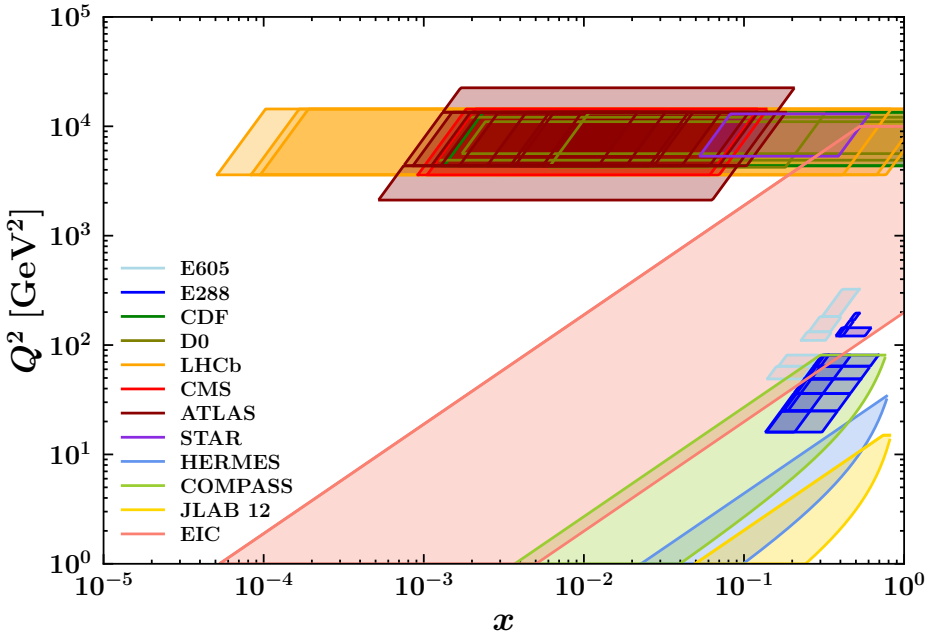


Figure 5.1: Kinematic coverage on the x vs. Q^2 plane of the some of the present and future (JLAB 12 and EIC) datasets for TMD extractions.

from Jefferson Lab at 12 GeV and from the Electron–Ion Collider with $\sqrt{s} = 141$ GeV.

On top of providing access to TMD fragmentation functions, we expect that the inclusion of Semi–Inclusive DIS data will have an impact on the determination of the x dependence of TMD PDFs and will make it possible to determine the flavor dependence of the nonperturbative function $f_{\rm NP}$. We remark that a better knowledge of TMDs will be important not only to obtain a deeper knowledge of hadron structure and QCD, but also for precision studies in high–energy processes involving hadrons, for instance for the determination of critical Standard Model parameters such as the W mass [88, 141].

In the future, the Electron–Ion Collider (EIC) will provide an unprecedented opportunity to make progress in the determination of TMDs [94, 219]. The US Department of Energy officially started the EIC project by establishing in December 2019. The realization of the EIC, which will be led jointly by Brookhaven National Laboratory and Thomas Jefferson National Accelerator Facility, is expected to take roughly a decade, with beam operations starting in the early 2030s.

The EIC will be capable of colliding beams of polarized electrons with polarized beams of light ions at an intensity, resolution and versatility never achieved before by an experimental apparatus [96].

Even though data from the EIC are to be expected after the late 2030s, the impact that they will have on the imaging of the nucleon in momentum space can be estimated using pseudodata based on `PYTHIA` simulations [220] and performing impact studies like

the ones illustrated in the EIC Yellow Report [96] and in references therein.

TMDs at the EIC will be accessed through Semi–Inclusive DIS and the studies presented in this thesis, as well as future developments of our numerical framework `NangaParbat`, will be very useful in the analysis of future data.

Acknowledgements

I am deeply grateful to my supervisor, Alessandro Bacchetta, for his guidance throughout my PhD. Alessandro, I truly appreciate the trust you placed in me right from the start and the useful advice you gave me throughout these years: no matter how busy you were, you always made time for me. Your intuitive approach to complex problems always inspired me, thank you for teaching me how to enjoy research and how to become a scientist. Also, thank you for giving me the opportunity to work on interesting projects and to attend national and international PhD schools, conferences and workshops.

I would also like to thank the referees of this thesis, Marc Schlegel and Emanuele Nocera: thank you for reading and commenting this work, your input was incredibly valuable and allowed me to improve the final result.

Part of my PhD was also supported by the Jefferson Laboratory EIC Fellowship: I am very grateful to Douglas Higinbotham and to all the members of the JLab EIC Center for this extraordinary opportunity. I would also like to thank Jianwei Qiu, who made my visiting time at the JLab Theory Center possible and who welcomed me there.

I would like to express my gratitude to all the members of our research group, as they provided a very pleasant and stimulating working environment as well as a wonderful company during the several social events, lunches and dinners that we shared during these years.

Thank you Barbara Pasquini, Marco Radici and Giuseppe Bozzi, talking with you was always insightful, and thank you Valerio Bertone, I did learn a lot while working with you. I would also like to thank Filippo Delcarro, first colleague PhD student in Pavia and then precious collaborator at JLab, Astrid Hiller–Blin, the best JLab office mate and fellow of hiking adventures I could have met.

A big thank you to Andrea Signori: you and Carlotta welcomed me in Newport News during my visiting time at JLab and made me feel at home. Andrea, I really enjoyed working with you, thanks for patiently answering all my questions and for

always asking if everything was fine.

I would like to say thank you also to Fulvio, my office mate for over two years: your sincere friendship and support have been very important for me during these years. Francesco Giovanni, your enthusiasm for research is contagious, thank you for encouraging me during these years and for being a great and special friend. Simone, thank you for being my travel companion for the fist times we went to the US, we shared marvelous experiences.

I would also like to thank the Collegio Nuovo, my home for all the years I studied in Pavia. On a more personal level, I would like to thank my dearest friends, Linda, Sara, Jude, Giulia and Giulia: thank you for all the dinners, the trips, the holidays and the life experiences that we shared. Even if we are scattered around the world, I have always felt you close.

Infine, vorrei dire grazie alla mia famiglia, a mia mamma, a mio papà e a mia sorella Laura, per l'affetto ed il supporto che mi avete sempre dimostrato.

List of publications

- A. Bacchetta, V. Bertone, C. Bissolotti, G. Bozzi, F. Delcarro, F. Piacenza et al., *Transverse–momentum–dependent parton distributions up to N^3LL from Drell–Yan data*, JHEP **07** (2020) 117 [arXiv:1912.07550].
- R. A. Khalek, A. Accardi, ..., C.Bissolotti et al., *Science Requirements and Detector Concepts for the Electron–Ion Collider: EIC Yellow Report*, [arXiv:2103.05419].
- N. A. Abdulov, A. Bacchetta, ..., C. Bissolotti et al., *TMDlib2 and TMDplotter: a platform for 3D hadron structure studies*, [arXiv:2103.09741].

Bibliography

[1] M. Gell-Mann, *A Schematic Model of Baryons and Mesons*, *Phys. Lett.* **8** (1964) 214.

[2] G. Zweig, *An SU(3) model for strong interaction symmetry and its breaking. Version 1*, .

[3] SLAC-MIT-Cal Tech collaboration, *Proposals for initial electron scattering experiments using the SLAC spectrometer facilities*, .

[4] E. D. Bloom et al., *High-Energy Inelastic e p Scattering at 6-Degrees and 10-Degrees*, *Phys. Rev. Lett.* **23** (1969) 930.

[5] M. Breidenbach, J. I. Friedman, H. W. Kendall, E. D. Bloom, D. H. Coward, H. C. DeStaebler et al., *Observed behavior of highly inelastic electron-proton scattering*, *Phys. Rev. Lett.* **23** (1969) 935.

[6] J. Bjorken, *Asymptotic Sum Rules at Infinite Momentum*, *Phys. Rev.* **179** (1969) 1547.

[7] R. P. Feynman, *Very high-energy collisions of hadrons*, *Phys. Rev. Lett.* **23** (1969) 1415.

[8] A. Zee, *Study of the renormalization group for small coupling constants*, *Phys. Rev. D* **7** (1973) 3630.

[9] H. Fritzsch, M. Gell-Mann and H. Leutwyler, *Advantages of the Color Octet Gluon Picture*, *Phys. Lett. B* **47** (1973) 365.

[10] D. Gross, *The discovery of asymptotic freedom and the emergence of QCD*, *Proc. Nat. Acad. Sci.* **102** (2005) 9099.

[11] K. G. Wilson, *Nonlagrangian models of current algebra*, *Phys. Rev.* **179** (1969) 1499.

[12] European Muon collaboration, *A Detailed Study of the Proton Structure Functions in Deep Inelastic Muon - Proton Scattering*, *Nucl. Phys. B* **259** (1985) 189.

[13] European Muon collaboration, *The Vertex and Large Angle Detectors of a Spectrometer System for High-energy Muon Physics*, *Nucl. Instrum. Meth.* **212** (1983) 111.

[14] D. Bollini et al., *A High Luminosity Spectrometer for Deep Inelastic Muon Scattering Experiments*, *Nucl. Instrum. Meth.* **204** (1983) 333.

[15] Bologna-CERN-Dubna-Munich-Saclay collaboration, *An Upgraded Configuration of a High Luminosity Spectrometer for Deep Inelastic Muon Scattering Experiments*, *Nucl. Instrum. Meth. A* **226** (1984) 330.

[16] G. K. Mallot and R. Voss, *Deep inelastic scattering with the SPS muon beam*, *Adv. Ser. Direct. High Energy Phys.* **23** (2015) 287.

[17] A. Zee, *Electron-positron annihilation in stagnant field theories*, *Phys. Rev. D* **8** (1973) 4038.

[18] G. Altarelli and G. Parisi, *Asymptotic Freedom in Parton Language*, *Nucl. Phys. B* **126** (1977) 298.

[19] L. Baulieu and C. Kounnas, *Equivalence of the 'Wilson Expansion' and a Generalized Parton Model*, *Nucl. Phys. B* **141** (1978) 423.

[20] J. Kodaira and T. Uematsu, *How Can We Define Properly the Q^2 Dependent Parton Distribution Function in {QCD}?*, *Nucl. Phys. B* **141** (1978) 497.

[21] T. Uematsu, *Q**2 Dependence of Quark and Gluon Fragmentation Functions in a Parton Picture Based on QCD*, *Phys. Lett. B* **79** (1978) 97.

[22] J. Owens, *On the Q**2 Dependence of Parton Fragmentation Functions*, *Phys. Lett. B* **76** (1978) 85.

[23] Y. L. Dokshitzer, *Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics.*, *Sov. Phys. JETP* **46** (1977) 641.

[24] V. Gribov and L. Lipatov, *Deep inelastic e p scattering in perturbation theory*, *Sov. J. Nucl. Phys.* **15** (1972) 438.

[25] W. A. Bardeen, A. Buras, D. Duke and T. Muta, *Deep Inelastic Scattering Beyond the Leading Order in Asymptotically Free Gauge Theories*, *Phys. Rev. D* **18** (1978) 3998.

[26] E. Floratos, D. Ross and C. T. Sachrajda, *Higher Order Effects in Asymptotically Free Gauge Theories: The Anomalous Dimensions of Wilson Operators*, *Nucl. Phys. B* **129** (1977) 66.

[27] E. Floratos, D. Ross and C. T. Sachrajda, *Higher Order Effects in Asymptotically Free Gauge Theories. 2. Flavor Singlet Wilson Operators and Coefficient Functions*, *Nucl. Phys. B* **152** (1979) 493.

[28] M. B. Einhorn and B. G. Weeks, *Jet Production by a Gluon Source in {QCD}*, *Nucl. Phys. B* **146** (1978) 445.

[29] G. F. Sterman and S. Weinberg, *Jets from Quantum Chromodynamics*, *Phys. Rev. Lett.* **39** (1977) 1436.

[30] K. Shizuya and S.-H. Tye, *Gluon Jets from Quantum Chromodynamics*, *Phys. Rev. Lett.* **41** (1978) 787.

[31] R. Ellis, H. Georgi, M. Machacek, H. Politzer and G. G. Ross, *Factorization and the Parton Model in QCD*, *Phys. Lett. B* **78** (1978) 281.

[32] A. H. Mueller, *Cut Vertices and their Renormalization: A Generalization of the Wilson Expansion*, *Phys. Rev. D* **18** (1978) 3705.

[33] D. Amati, R. Petronzio and G. Veneziano, *Relating Hard QCD Processes Through Universality of Mass Singularities. 2.*, *Nucl. Phys. B* **146** (1978) 29.

[34] D. Amati, R. Petronzio and G. Veneziano, *Relating Hard QCD Processes Through Universality of Mass Singularities*, *Nucl. Phys. B* **140** (1978) 54.

[35] W. Celmaster and R. J. Gonsalves, *An Analytic Calculation of Higher Order Quantum Chromodynamic Corrections in e+ e- Annihilation*, *Phys. Rev. Lett.* **44** (1980) 560.

[36] W. Celmaster and R. J. Gonsalves, *Fourth Order QCD Contributions to the e+ e- Annihilation Cross-Section*, *Phys. Rev. D* **21** (1980) 3112.

[37] K. Chetyrkin, A. Kataev and F. Tkachov, *Higher Order Corrections to Sigma-t (e+ e- —> Hadrons) in Quantum Chromodynamics*, *Phys. Lett. B* **85** (1979) 277.

[38] K. Chetyrkin, A. Kataev and F. Tkachov, *New Approach to Evaluation of Multiloop Feynman Integrals: The Gegenbauer Polynomial x Space Technique*, *Nucl. Phys. B* **174** (1980) 345.

[39] M. Dine and J. Sapirstein, *Higher Order QCD Corrections in e+ e- Annihilation*, *Phys. Rev. Lett.* **43** (1979) 668.

[40] W. A. Bardeen and A. J. Buras, *Higher Order Asymptotic Freedom Corrections to Photon - Photon Scattering*, *Phys. Rev. D* **20** (1979) 166.

[41] R. Ellis, D. Ross and A. Terrano, *Calculation of Event Shape Parameters in e+ e- Annihilation*, *Phys. Rev. Lett.* **45** (1980) 1226.

[42] R. Ellis, D. Ross and A. Terrano, *The Perturbative Calculation of Jet Structure in e+ e- Annihilation*, *Nucl. Phys. B* **178** (1981) 421.

[43] J. C. Collins, D. E. Soper and G. F. Sterman, *Factorization of Hard Processes in QCD*, *Adv. Ser. Direct. High Energy Phys.* **5** (1989) 1 [arXiv:hep-ph/0409313].

[44] J. Collins, *Foundations of perturbative QCD*, *Camb. Monogr. Part. Phys. Nucl. Phys. Cosmol.* **32** (2011) 1.

[45] CTEQ collaboration, *Handbook of perturbative QCD: Version 1.0*, *Rev. Mod. Phys.* **67** (1995) 157.

[46] S. D. Drell and T.-M. Yan, *Massive Lepton Pair Production in Hadron-Hadron Collisions at High-Energies*, *Phys. Rev. Lett.* **25** (1970) 316.

[47] G. Altarelli, R. Ellis and G. Martinelli, *Leptoproduction and Drell-Yan Processes Beyond the Leading Approximation in Chromodynamics*, *Nucl. Phys. B* **143** (1978) 521.

[48] W. Furmanski and R. Petronzio, *Singlet Parton Densities Beyond Leading Order*, *Phys. Lett. B* **97** (1980) 437.

[49] G. Curci, W. Furmanski and R. Petronzio, *Evolution of Parton Densities Beyond Leading Order: The Nonsinglet Case*, *Nucl. Phys. B* **175** (1980) 27.

[50] S. Forte and G. Watt, *Progress in the Determination of the Partonic Structure of the Proton*, *Ann. Rev. Nucl. Part. Sci.* **63** (2013) 291 [arXiv:1301.6754].

[51] J. J. Ethier and E. R. Nocera, *Parton Distributions in Nucleons and Nuclei*, *Ann. Rev. Nucl. Part. Sci.* **70** (2020) 43 [arXiv:2001.07722].

[52] K. Kovařík, P. M. Nadolsky and D. E. Soper, *Hadronic structure in high-energy collisions*, *Rev. Mod. Phys.* **92** (2020) 045003 [arXiv:1905.06957].

[53] T. C. Rogers, *An overview of transverse-momentum–dependent factorization and evolution*, *Eur. Phys. J.* **A52** (2016) 153 [arXiv:1509.04766].

[54] M. Diehl, *Introduction to GPDs and TMDs*, *Eur. Phys. J.* **A52** (2016) 149 [arXiv:1512.01328].

[55] R. Angeles-Martinez et al., *Transverse Momentum Dependent (TMD) parton distribution functions: status and prospects*, *Acta Phys. Polon.* **B46** (2015) 2501 [arXiv:1507.05267].

[56] P. J. Mulders and R. D. Tangerman, *The Complete tree level result up to order 1/Q for polarized deep inelastic leptoproduction*, *Nucl. Phys.* **B461** (1996) 197 [arXiv:hep-ph/9510301].

[57] D. Boer and P. J. Mulders, *Time reversal odd distribution functions in leptoproduction*, *Phys. Rev.* **D57** (1998) 5780 [arXiv:hep-ph/9711485].

[58] A. Bacchetta, M. Diehl, K. Goeke, A. Metz, P. J. Mulders and M. Schlegel, *Semi-inclusive deep inelastic scattering at small transverse momentum*, *JHEP* **02** (2007) 093 [arXiv:hep-ph/0611265].

[59] A. Metz, M. Schlegel and K. Goeke, *Transverse single spin asymmetries in inclusive deep-inelastic scattering*, *Phys. Lett. B* **643** (2006) 319 [arXiv:hep-ph/0610112].

[60] R. Tangerman and P. Mulders, *Polarized twist - three distributions g(T) and h(L) and the role of intrinsic transverse momentum*, [arXiv:hep-ph/9408305].

[61] R. Tangerman and P. Mulders, *Intrinsic transverse momentum and the polarized Drell-Yan process*, *Phys. Rev. D* **51** (1995) 3357 [arXiv:hep-ph/9403227].

[62] J. P. Ralston and D. E. Soper, *Production of Dimuons from High-Energy Polarized Proton Proton Collisions*, *Nucl. Phys. B* **152** (1979) 109.

[63] A. Accardi and A. Bacchetta, *Accessing the nucleon transverse structure in inclusive deep inelastic scattering*, *Phys. Lett. B* **773** (2017) 632 [arXiv:1706.02000].

[64] G. A. Miller, *Shapes of the proton*, *Phys. Rev. C* **68** (2003) 022201 [arXiv:nucl-th/0304076].

[65] D. W. Sivers, *Single Spin Production Asymmetries from the Hard Scattering of Point-Like Constituents*, *Phys. Rev. D* **41** (1990) 83.

[66] A. Kotzinian and P. Mulders, *Probing transverse quark polarization via azimuthal asymmetries in leptoproduction*, *Phys. Lett. B* **406** (1997) 373 [arXiv:hep-ph/9701330].

[67] C. T. H. Davies, B. R. Webber and W. J. Stirling, *Drell-Yan Cross-Sections at Small Transverse Momentum*, *Nucl. Phys.* **B256** (1985) 413.

[68] G. A. Ladinsky and C. P. Yuan, *The Nonperturbative regime in QCD resummation for gauge boson production at hadron colliders*, *Phys. Rev.* **D50** (1994) R4239 [arXiv:hep-ph/9311341].

[69] F. Landry, R. Brock, G. Ladinsky and C. P. Yuan, *New fits for the nonperturbative parameters in the CSS resummation formalism*, *Phys. Rev.* **D63** (2001) 013004 [arXiv:hep-ph/9905391].

[70] J.-w. Qiu and X.-f. Zhang, *Role of the nonperturbative input in QCD resummed Drell-Yan Q_T distributions*, *Phys. Rev.* **D63** (2001) 114011 [arXiv:hep-ph/0012348].

[71] F. Landry, R. Brock, P. M. Nadolsky and C. P. Yuan, *Tevatron Run-1 Z boson data and Collins-Soper-Sterman resummation formalism*, *Phys. Rev.* **D67** (2003) 073016 [arXiv:hep-ph/0212159].

[72] A. V. Konychev and P. M. Nadolsky, *Universality of the Collins-Soper-Sterman nonperturbative function in gauge boson production*, *Phys. Lett.* **B633** (2006) 710 [arXiv:hep-ph/0506225].

[73] T. Becher, M. Neubert and D. Wilhelm, *Electroweak Gauge-Boson Production at Small q_T: Infrared Safety from the Collinear Anomaly*, *JHEP* **02** (2012) 124 [arXiv:1109.6027].

[74] S. Camarda et al., *DYTurbo: Fast predictions for Drell-Yan processes*, [arXiv:1910.07049].

[75] R. Meng, F. I. Olness and D. E. Soper, *Semiinclusive deeply inelastic scattering at small q(T)*, *Phys. Rev.* **D54** (1996) 1919 [arXiv:hep-ph/9511311].

[76] P. M. Nadolsky, D. R. Stump and C. P. Yuan, *Semiinclusive hadron production at HERA: The Effect of QCD gluon resummation*, *Phys. Rev.* **D61** (2000) 014003 [arXiv:hep-ph/9906280].

[77] U. D'Alesio, M. G. Echevarria, S. Melis and I. Scimemi, *Non-perturbative QCD effects in q_T spectra of Drell-Yan and Z-boson production*, *JHEP* **11** (2014) 098 [arXiv:1407.3311].

[78] I. Scimemi and A. Vladimirov, *Analysis of vector boson production within TMD factorization*, *Eur. Phys. J.* **C78** (2018) 89 [arXiv:1706.01473].

[79] V. Bertone, I. Scimemi and A. Vladimirov, *Extraction of unpolarized quark transverse momentum dependent parton distributions from Drell-Yan/Z-boson production*, *JHEP* **06** (2019) 028 [arXiv:1902.08474].

[80] A. Signori, A. Bacchetta, M. Radici and G. Schnell, *Investigations into the flavor dependence of partonic transverse momentum*, *JHEP* **11** (2013) 194 [arXiv:1309.3507].

[81] M. Anselmino, M. Boglione, J. O. Gonzalez Hernandez, S. Melis and A. Prokudin, *Unpolarised Transverse Momentum Dependent Distribution and Fragmentation Functions from SIDIS Multiplicities*, *JHEP* **04** (2014) 005 [arXiv:1312.6261].

[82] M. G. Echevarria, A. Idilbi, Z.-B. Kang and I. Vitev, *QCD Evolution of the Sivers Asymmetry*, *Phys. Rev.* **D89** (2014) 074013 [arXiv:1401.5078].

[83] P. Sun, J. Isaacson, C. P. Yuan and F. Yuan, *Nonperturbative functions for SIDIS and Drell–Yan processes*, *Int. J. Mod. Phys.* **A33** (2018) 1841006 [arXiv:1406.3073].

[84] A. Bacchetta, F. Delcarro, C. Pisano, M. Radici and A. Signori, *Extraction of partonic transverse momentum distributions from semi-inclusive deep-inelastic scattering, Drell-Yan and Z-boson production*, *JHEP* **06** (2017) 081 [arXiv:1703.10157].

[85] I. Scimemi and A. Vladimirov, *Non-perturbative structure of semi-inclusive deep-inelastic and Drell-Yan scattering at small transverse momentum*, [arXiv:1912.06532].

[86] A. Bermudez Martinez, P. Connor, H. Jung, A. Lelek, R. Žlebčík, F. Hautmann et al., *Collinear and TMD parton densities from fits to precision DIS measurements in the parton branching method*, *Phys. Rev.* **D99** (2019) 074008 [arXiv:1804.11152].

[87] A. Bermudez Martinez et al., *Production of Z-bosons in the parton branching method*, *Phys. Rev.* **D100** (2019) 074027 [arXiv:1906.00919].

[88] A. Bacchetta, G. Bozzi, M. Radici, M. Ritzmann and A. Signori, *Effect of Flavor-Dependent Partonic Transverse Momentum on the Determination of the W Boson Mass in Hadronic Collisions*, *Phys. Lett.* **B788** (2019) 542 [arXiv:1807.02101].

[89] W. Bizon, P. F. Monni, E. Re, L. Rottoli and P. Torrielli, *Momentum-space resummation for transverse observables and the Higgs $p_\perp$ at $N^3LL+NNLO$*, *JHEP* **02** (2018) 108 [arXiv:1705.09127].

[90] W. Bizoń, X. Chen, A. Gehrmann-De Ridder, T. Gehrmann, N. Glover, A. Huss et al., *Fiducial distributions in Higgs and Drell-Yan production at $N^3LL+NNLO$*, *JHEP* **12** (2018) 132 [arXiv:1805.05916].

[91] T. Becher, M. Neubert and D. Wilhelm, *Higgs-Boson Production at Small Transverse Momentum*, *JHEP* **05** (2013) 110 [arXiv:1212.2621].

[92] F. Coradeschi and T. Cridge, *reSolve — A transverse momentum resummation tool*, *Comput. Phys. Commun.* **238** (2019) 262 [arXiv:1711.02083].

[93] N. A. of Sciences Engineering and Medicine, *An Assessment of U.S.-Based Electron-Ion Collider Science*. The National Academies Press, Washington, DC, 2018, 10.17226/25171.

[94] A. Accardi et al., *Electron Ion Collider: The Next QCD Frontier*, *Eur. Phys. J.* **A52** (2016) 268 [arXiv:1212.1701].

[95] A. Bacchetta, V. Bertone, C. Bissolotti, G. Bozzi, F. Delcarro, F. Piacenza et al., *Transverse-momentum-dependent parton distributions up to N^3LL from Drell-Yan data*, *JHEP* **07** (2020) 117 [arXiv:1912.07550].

[96] R. A. Khalek, A. Accardi, J. Adam, D. Adamiak, W. Akers, M. Albaladejo et al., *Science Requirements and Detector Concepts for the Electron-Ion Collider: EIC Yellow Report*, [arXiv:2103.05419].

[97] J. C. Collins and D. E. Soper, *Parton Distribution and Decay Functions*, *Nucl. Phys. B* **194** (1982) 445.

[98] J. C. Collins, D. E. Soper and G. F. Sterman, *Transverse Momentum Distribution in Drell-Yan Pair and W and Z Boson Production*, *Nucl. Phys.* **B250** (1985) 199.

[99] J. C. Collins and D. E. Soper, *Back-To-Back Jets in QCD*, *Nucl. Phys.* **B193** (1981) 381.

[100] S. M. Aybat and T. C. Rogers, *TMD Parton Distribution and Fragmentation Functions with QCD Evolution*, *Phys. Rev.* **D83** (2011) 114042 [arXiv:1101.5057].

[101] X.-d. Ji, J.-p. Ma and F. Yuan, *QCD factorization for semi-inclusive deep-inelastic scattering at low transverse momentum*, *Phys. Rev.* **D71** (2005) 034005 [arXiv:hep-ph/0404183].

[102] M. Anselmino, A. Mukherjee and A. Vossen, *Transverse spin effects in hard semi-inclusive collisions*, *Prog. Part. Nucl. Phys.* **114** (2020) 103806 [arXiv:2001.05415].

[103] J.-y. Chiu, A. Jain, D. Neill and I. Z. Rothstein, *The Rapidity Renormalization Group*, *Phys. Rev. Lett.* **108** (2012) 151601 [arXiv:1104.0881].

[104] J.-Y. Chiu, A. Jain, D. Neill and I. Z. Rothstein, *A Formalism for the Systematic Treatment of Rapidity Logarithms in Quantum Field Theory*, *JHEP* **05** (2012) 084 [arXiv:1202.0814].

[105] I. Scimemi and A. Vladimirov, *Systematic analysis of double-scale evolution*, *JHEP* **08** (2018) 003 [arXiv:1803.11089].

[106] G. Billis, F. J. Tackmann and J. Talbert, *Higher-Order Sudakov Resummation in Coupled Gauge Theories*, [arXiv:1907.02971].

[107] M.-x. Luo, T.-Z. Yang, H. X. Zhu and Y. J. Zhu, *Quark Transverse Parton Distribution at the Next-to-Next-to-Next-to-Leading Order*, *Phys. Rev. Lett.* **124** (2020) 092001 [arXiv:1912.05778].

[108] J. Collins and T. C. Rogers, *Connecting Different TMD Factorization Formalisms in QCD*, *Phys. Rev.* **D96** (2017) 054011 [arXiv:1705.07167].

[109] S. Catani, L. Cieri, D. de Florian, G. Ferrera and M. Grazzini, *Vector boson production at hadron colliders: hard-collinear coefficients at the NNLO*, *Eur. Phys. J.* **C72** (2012) 2195 [arXiv:1209.0158].

[110] M. G. Echevarria, I. Scimemi and A. Vladimirov, *Unpolarized Transverse Momentum Dependent Parton Distribution and Fragmentation Functions at next-to-next-to-leading order*, *JHEP* **09** (2016) 004 [arXiv:1604.07869].

[111] Y. Li and H. X. Zhu, *Bootstrapping Rapidity Anomalous Dimensions for Transverse-Momentum Resummation*, *Phys. Rev. Lett.* **118** (2017) 022004 [arXiv:1604.01404].

[112] J. Davies, A. Vogt, B. Ruijl, T. Ueda and J. A. M. Vermaseren, *Large-n_f contributions to the four-loop splitting functions in QCD*, *Nucl. Phys.* **B915** (2017) 335 [arXiv:1610.07477].

[113] S. Moch, B. Ruijl, T. Ueda, J. A. M. Vermaseren and A. Vogt, *Four-Loop Non-Singlet Splitting Functions in the Planar Limit and Beyond*, *JHEP* **10** (2017) 041 [arXiv:1707.08315].

[114] S. Moch, B. Ruijl, T. Ueda, J. A. M. Vermaseren and A. Vogt, *On quartic colour factors in splitting functions and the gluon cusp anomalous dimension*, *Phys. Lett.* **B782** (2018) 627 [arXiv:1805.09638].

[115] I. W. Stewart, F. J. Tackmann, J. R. Walsh and S. Zuberi, *Jet p_T resummation in Higgs production at $NNLL' + NNLO$*, *Phys. Rev.* **D89** (2014) 054001 [arXiv:1307.1808].

[116] M. A. Ebert and F. J. Tackmann, *Resummation of Transverse Momentum Distributions in Distribution Space*, *JHEP* **02** (2017) 110 [arXiv:1611.08610].

[117] S. Catani, *Higher order QCD corrections in hadron collisions: Soft gluon resummation and exponentiation*, *Nucl. Phys. Proc. Suppl.* **54A** (1997) 107 [arXiv:hep-ph/9610413].

[118] G. Bozzi, S. Catani, G. Ferrera, D. de Florian and M. Grazzini, *Production of Drell-Yan lepton pairs in hadron collisions: Transverse-momentum resummation at next-to-next-to-leading logarithmic accuracy*, *Phys. Lett.* **B696** (2011) 207 [arXiv:1007.2351].

[119] A. Banfi, H. McAslan, P. F. Monni and G. Zanderighi, *The two-jet rate in e^+e^- at next-to-next-to-leading-logarithmic order*, *Phys. Rev. Lett.* **117** (2016) 172001 [arXiv:1607.03111].

[120] G. Bozzi, S. Catani, D. de Florian and M. Grazzini, *Transverse-momentum resummation and the spectrum of the Higgs boson at the LHC*, *Nucl. Phys.* **B737** (2006) 73 [arXiv:hep-ph/0508068].

[121] S. Catani, L. Cieri, D. de Florian, G. Ferrera and M. Grazzini, *Universality of transverse-momentum resummation and hard factors at the NNLO*, *Nucl. Phys.* **B881** (2014) 414 [arXiv:1311.1654].

[122] C. Muselli, S. Forte and G. Ridolfi, *Combined threshold and transverse momentum resummation for inclusive observables*, *JHEP* **03** (2017) 106 [arXiv:1701.01464].

[123] S. Alioli, A. Broggio, S. Kallweit, M. A. Lim and L. Rottoli, *Higgsstrahlung at NNLL'+NNLO matched to parton showers in GENEVA*, *Phys. Rev.* **D100** (2019) 096016 [arXiv:1909.02026].

[124] M. G. Echevarria, T. Kasemets, J.-P. Lansberg, C. Pisano and A. Signori, *Matching factorization theorems with an inverse-error weighting*, *Phys. Lett.* **B781** (2018) 161 [arXiv:1801.01480].

[125] G. Lustermans, J. K. L. Michel, F. J. Tackmann and W. J. Waalewijn, *Joint two-dimensional resummation in q_T and 0-jettiness at NNLL*, *JHEP* **03** (2019) 124 [arXiv:1901.03331].

[126] L. A. Harland-Lang, A. D. Martin, P. Motylinski and R. S. Thorne, *Parton distributions in the LHC era: MMHT 2014 PDFs*, *Eur. Phys. J.* **C75** (2015) 204 [arXiv:1412.3989].

[127] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht et al., *LHAPDF6: parton density access in the LHC precision era*, *Eur. Phys. J.* **C75** (2015) 132 [arXiv:1412.7420].

[128] S. Catani, D. de Florian and M. Grazzini, *Universality of nonleading logarithmic contributions in transverse momentum distributions*, *Nucl. Phys. B* **596** (2001) 299 [arXiv:hep-ph/0008184].

[129] G. Bozzi, S. Catani, G. Ferrera, D. de Florian and M. Grazzini, *Transverse-momentum resummation: A Perturbative study of Z production at the Tevatron*, *Nucl. Phys. B* **815** (2009) 174 [arXiv:0812.2862].

[130] S. Catani, D. de Florian, G. Ferrera and M. Grazzini, *Vector boson production at hadron colliders: transverse-momentum resummation and leptonic decay*, *JHEP* **12** (2015) 047 [arXiv:1507.06937].

[131] F. Hautmann, I. Scimemi and A. Vladimirov, *Non-perturbative contributions to vector-boson transverse momentum spectra in hadronic collisions*, *Phys. Lett. B* **806** (2020) 135478 [arXiv:2002.12810].

[132] S. Catani, M. L. Mangano, P. Nason and L. Trentadue, *The Resummation of soft gluons in hadronic collisions*, *Nucl. Phys.* **B478** (1996) 273 [arXiv:hep-ph/9604351].

[133] J. Collins, L. Gamberg, A. Prokudin, T. C. Rogers, N. Sato and B. Wang, *Relating Transverse Momentum Dependent and Collinear Factorization Theorems in a Generalized Formalism*, *Phys. Rev.* **D94** (2016) 034014 [arXiv:1605.00671].

[134] J. Collins and T. Rogers, *Understanding the large-distance behavior of transverse-momentum-dependent parton densities and the Collins-Soper evolution kernel*, *Phys. Rev.* **D91** (2015) 074020 [arXiv:1412.3820].

[135] J.-w. Qiu and X.-f. Zhang, *QCD prediction for heavy boson transverse momentum distributions*, *Phys. Rev. Lett.* **86** (2001) 2724 [arXiv:hep-ph/0012058].

[136] J.-w. Qiu and X.-f. Zhang, *Role of nonperturbative input in QCD resummed heavy boson Q(T) distribution*, in *37th Rencontres de Moriond on QCD and Hadronic Interactions*, pp. 169–172, 5, 2002, hep-ph/0205115.

[137] M. Burkardt and B. Pasquini, *Modelling the nucleon structure*, *Eur. Phys. J.* **A52** (2016) 161 [arXiv:1510.02567].

[138] D. Müller and D. S. Hwang, *The concept of phenomenological light-front wave functions – Regge improved diquark model predictions*, [arXiv:1407.1655].

[139] HERMES collaboration, *Multiplicities of charged pions and kaons from semi-inclusive deep-inelastic scattering by the proton and the deuteron*, *Phys. Rev.* **D87** (2013) 074029 [arXiv:1212.5407].

[140] COMPASS collaboration, *Transverse-momentum-dependent Multiplicities of Charged Hadrons in Muon-Deuteron Deep Inelastic Scattering*, *Phys. Rev.* **D97** (2018) 032006 [arXiv:1709.07374].

[141] G. Bozzi and A. Signori, *Nonperturbative Uncertainties on the Transverse Momentum Distribution of Electroweak Bosons and on the Determination of the Boson Mass at the LHC*, *Adv. High Energy Phys.* **2019** (2019) 2526897 [arXiv:1901.01162].

[142] A. Bacchetta, F. Conti and M. Radici, *Transverse-momentum distributions in a diquark spectator model*, *Phys. Rev.* **D78** (2008) 074010 [arXiv:0807.0323].

[143] M. Wakamatsu, *Transverse momentum distributions of quarks in the nucleon from the Chiral Quark Soliton Model*, *Phys. Rev.* **D79** (2009) 094028 [arXiv:0903.1886].

[144] A. V. Efremov, P. Schweitzer, O. V. Teryaev and P. Zavada, *The relation between TMDs and PDFs in the covariant parton model approach*, *Phys. Rev.* **D83** (2011) 054025 [arXiv:1012.5296].

[145] C. Bourrely, F. Buccella and J. Soffer, *Semiinclusive DIS cross sections and spin asymmetries in the quantum statistical parton distributions approach*, *Phys. Rev.* **D83** (2011) 074008 [arXiv:1008.5322].

[146] H. H. Matevosyan, W. Bentz, I. C. Cloet and A. W. Thomas, *Transverse Momentum Dependent Fragmentation and Quark Distribution Functions from the NJL-jet Model*, *Phys. Rev.* **D85** (2012) 014021 [arXiv:1111.1740].

[147] P. Schweitzer, M. Strikman and C. Weiss, *Intrinsic transverse momentum and parton correlations from dynamical chiral symmetry breaking*, *JHEP* **01** (2013) 163 [arXiv:1210.1267].

[148] B. U. Musch, P. Hagler, J. W. Negele and A. Schafer, *Exploring quark transverse momentum distributions with lattice QCD*, *Phys. Rev.* **D83** (2011) 094507 [arXiv:1011.1213].

[149] A. Bacchetta and M. G. Echevarria, *QCD×QED evolution of TMDs*, *Phys. Lett.* **B788** (2019) 280 [arXiv:1810.02297].

[150] L. Cieri, G. Ferrera and G. F. R. Sborlini, *Combining QED and QCD transverse-momentum resummation for Z boson production at hadron colliders*, *JHEP* **08** (2018) 165 [arXiv:1805.11948].

[151] O. Lupton and M. Vesterinen, *Simultaneously determining the $W^{\pm}$ boson mass and p_{T} spectrum*, [arXiv:1907.09958].

[152] R.-b. Meng, F. I. Olness and D. E. Soper, *Semiinclusive deeply inelastic scattering at electron - proton colliders*, *Nucl. Phys.* **B371** (1992) 79.

[153] C. A. Aidala, B. Field, L. P. Gamberg and T. C. Rogers, *Limits on transverse momentum dependent evolution from semi-inclusive deep inelastic scattering at moderate Q*, *Phys. Rev.* **D89** (2014) 094002 [arXiv:1401.2654].

[154] Z.-B. Kang, A. Prokudin, P. Sun and F. Yuan, *Extraction of Quark Transversity Distribution and Collins Fragmentation Functions with QCD Evolution*, *Phys. Rev.* **D93** (2016) 014009 [arXiv:1505.05589].

[155] M. A. Ebert, I. W. Stewart and Y. Zhao, *Determining the Nonperturbative Collins-Soper Kernel From Lattice QCD*, *Phys. Rev.* **D99** (2019) 034505 [arXiv:1811.00026].

[156] H. H. Matevosyan, W. Bentz, I. C. Cloët and A. W. Thomas, *Transverse-momentum-dependent fragmentation and quark distribution functions from the nambu–jona-lasinio–jet model*, *Phys. Rev. D* **85** (2012) 014021.

[157] M. Anselmino, M. Boglione, U. D'Alesio, A. Kotzinian, F. Murgia and A. Prokudin, *Role of cahn and sivers effects in deep inelastic scattering*, *Phys. Rev. D* **71** (2005) 074006.

[158] A. Bacchetta, D. Boer, M. Diehl and P. J. Mulders, *Matches and mismatches in the descriptions of semi-inclusive processes at low and high transverse momentum*, *JHEP* **08** (2008) 023 [arXiv:0803.0227].

[159] X.-d. Ji and F. Yuan, *Parton distributions in light cone gauge: Where are the final state interactions?*, *Phys. Lett.* **B543** (2002) 66 [arXiv:hep-ph/0206057].

[160] M. G. Echevarria, A. Idilbi, A. Schäfer and I. Scimemi, *Model-Independent Evolution of Transverse Momentum Dependent Distribution Functions (TMDs) at NNLL*, *Eur. Phys. J.* **C73** (2013) 2636 [arXiv:1208.1281].

[161] J. C. Collins and T. C. Rogers, *Equality of Two Definitions for Transverse Momentum Dependent Parton Distribution Functions*, *Phys. Rev.* **D87** (2013) 034018 [arXiv:1210.2100].

[162] I. Scimemi and A. Vladimirov, *Power corrections and renormalons in Transverse Momentum Distributions*, *JHEP* **03** (2017) 002 [arXiv:1609.06047].

[163] M. Gluck, P. Jimenez-Delgado and E. Reya, *Dynamical parton distributions of the nucleon and very small-x physics*, *Eur. Phys. J.* **C53** (2008) 355 [arXiv:0709.0614].

[164] D. de Florian, R. Sassot, M. Epele, R. J. Hernández-Pinto and M. Stratmann, *Parton-to-Pion Fragmentation Reloaded*, *Phys. Rev.* **D91** (2015) 014035 [arXiv:1410.6027].

[165] D. de Florian, R. Sassot and M. Stratmann, *Global analysis of fragmentation functions for pions and kaons and their uncertainties*, *Phys. Rev.* **D75** (2007) 114010 [arXiv:hep-ph/0703242].

[166] COMPASS collaboration, *Hadron Transverse Momentum Distributions in Muon Deep Inelastic Scattering at 160 GeV/c*, *Eur. Phys. J.* **C73** (2013) 2531 [arXiv:1305.7317].

[167] ZEUS collaboration, *Measurement of the Longitudinal Proton Structure Function at HERA*, *Phys. Lett.* **B682** (2009) 8 [arXiv:0904.1092].

[168] H1 collaboration, *Measurement of inclusive ep cross sections at high Q^2 at $\sqrt{s} = 225$ and 252 GeV and of the longitudinal proton structure function F_L at HERA*, *Eur. Phys. J.* **C74** (2014) 2814 [arXiv:1312.4821].

[169] NNPDF collaboration, *A Determination of parton distributions with faithful uncertainty estimation*, *Nucl. Phys.* **B809** (2009) 1 [arXiv:0808.1231].

[170] T. Kluge, K. Rabbertz and M. Wobisch, *FastNLO: Fast pQCD calculations for PDF fits*, in *Deep inelastic scattering. Proceedings, 14th International Workshop, DIS 2006, Tsukuba, Japan, April 20-24, 2006*, pp. 483–486, 2006, hep-ph/0609285, DOI.

[171] fastNLO collaboration, *New features in version 2 of the fastNLO project*, in *Proceedings, 20th International Workshop on Deep-Inelastic Scattering and Related Subjects (DIS 2012): Bonn, Germany, March 26-30, 2012*, pp. 217–221, 2012, 1208.3641, DOI.

[172] T. Carli, D. Clements, A. Cooper-Sarkar, C. Gwenlan, G. P. Salam, F. Siegert et al., *A posteriori inclusion of parton density functions in NLO QCD final-state calculations at hadron colliders: The APPLGRID Project*, *Eur. Phys. J.* **C66** (2010) 503 [arXiv:0911.2985].

[173] M. Wobisch, *Measurement and QCD analysis of jet cross-sections in deep inelastic positron proton collisions at $\sqrt{s} = 300$ GeV*, Master's thesis, 2000.

[174] S. Carrazza, E. R. Nocera, C. Schwan and M. Zaro, *PineAPPL: combining EW and QCD corrections for fast evaluation of LHC processes*, *JHEP* **12** (2020) 108 [arXiv:2008.12789].

[175] fastNLO collaboration, *Theory-Data Comparisons for Jet Measurements in Hadron-Induced Processes*, [arXiv:1109.1310].

[176] V. Bertone, S. Carrazza and N. P. Hartland, *APFELgrid: a high performance tool for parton density determinations*, *Comput. Phys. Commun.* **212** (2017) 205 [arXiv:1605.02070].

[177] A. D. Martin, W. J. Stirling, R. S. Thorne and G. Watt, *Parton distributions for the LHC*, *Eur. Phys. J.* **C63** (2009) 189 [arXiv:0901.0002].

[178] H.-L. Lai, M. Guzzi, J. Huston, Z. Li, P. M. Nadolsky, J. Pumplin et al., *New parton distributions for collider physics*, *Phys. Rev. D* **82** (2010) 074024 [arXiv:1007.2241].

[179] R. D. Ball, V. Bertone, F. Cerutti, L. Del Debbio, S. Forte, A. Guffanti et al., *Impact of Heavy Quark Masses on Parton Distributions and LHC Phenomenology*, *Nucl. Phys. B* **849** (2011) 296 [arXiv:1101.1300].

[180] H. Ogata, *A numerical integration formula based on the bessel functions*, *Publications of the Research Institute for Mathematical Sciences* **41** (2005) 949.

[181] V. Bertone, S. Carrazza and J. Rojo, *APFEL: A PDF Evolution Library with QED corrections*, *Comput. Phys. Commun.* **185** (2014) 1647 [arXiv:1310.1394].

[182] V. Bertone, *APFEL++: A new PDF evolution library in C++*, *PoS* **DIS2017** (2018) 201 [arXiv:1708.00911].

[183] R. K. Ellis, W. J. Stirling and B. R. Webber, *QCD and collider physics*, vol. 8. Cambridge University Press, 2, 2011.

[184] G. D'Agostini, *Asymmetric uncertainties: Sources, treatment and potential dangers*, [arXiv:physics/0403086].

[185] G. D'Agostini, *On the use of the covariance matrix to fit correlated data*, *Nucl. Instrum. Meth.* **A346** (1994) 306.

[186] G. D'Agostini, *Bayesian reasoning in data analysis: A critical introduction.* 2003.

[187] NNPDF collaboration, *Fitting Parton Distribution Data with Multiplicative Normalization Uncertainties*, *JHEP* **05** (2010) 075 [arXiv:0912.2276].

[188] R. D. Ball et al., *Parton Distribution Benchmarking with LHC Data*, *JHEP* **04** (2013) 125 [arXiv:1211.5142].

[189] F. James, *MINUIT Function Minimization and Error Analysis: Reference Manual Version 94.1*, .

[190] S. Agarwal, K. Mierle and Others, "Ceres solver." http://ceres-solver.org.

[191] S. Alekhin et al., *HERAFitter*, *Eur. Phys. J.* **C75** (2015) 304 [arXiv:1410.4412].

[192] T. xFitter web page. https://www.xfitter.org/xFitter/.

[193] F. Hautmann, H. Jung, M. Krämer, P. J. Mulders, E. R. Nocera, T. C. Rogers et al., *TMDlib and TMDplotter: library and plotting tools for transverse-momentum-dependent parton distributions*, *Eur. Phys. J.* **C74** (2014) 3220 [arXiv:1408.3015].

[194] N. A. Abdulov et al., *TMDlib2 and TMDplotter: a platform for 3D hadron structure studies*, [arXiv:2103.09741].

[195] S. Carrazza, J. I. Latorre, J. Rojo and G. Watt, *A compression algorithm for the combination of PDF sets*, *Eur. Phys. J. C* **75** (2015) 474 [arXiv:1504.06469].

[196] D. Boer, *Investigating the origins of transverse spin asymmetries at RHIC*, *Phys. Rev.* **D60** (1999) 014012 [arXiv:hep-ph/9902255].

[197] A. Metz and A. Vossen, *Parton Fragmentation Functions*, *Prog. Part. Nucl. Phys.* **91** (2016) 136 [arXiv:1607.02521].

[198] J. Levelt and P. J. Mulders, *Quark correlation functions in deep inelastic semiinclusive processes*, *Phys. Rev. D* **49** (1994) 96 [arXiv:hep-ph/9304232].

[199] G. Moreno et al., *Dimuon production in proton - copper collisions at* $\sqrt{s}$ = *38.8-GeV*, *Phys. Rev.* **D43** (1991) 2815.

[200] A. S. Ito et al., *Measurement of the Continuum of Dimuons Produced in High-Energy Proton - Nucleus Collisions*, *Phys. Rev.* **D23** (1981) 604.

[201] CDF collaboration, *The transverse momentum and total cross section of* e^+e^- *pairs in the* Z *boson region from* $p\bar{p}$ *collisions at* $\sqrt{s} = 1.8$ *TeV*, *Phys. Rev. Lett.* **84** (2000) 845 [arXiv:hep-ex/0001021].

[202] CDF collaboration, *Transverse momentum cross section of* e^+e^- *pairs in the* Z*-boson region from* $p\bar{p}$ *collisions at* $\sqrt{s} = 1.96$ *TeV*, *Phys. Rev.* **D86** (2012) 052010 [arXiv:1207.7138].

[203] D0 collaboration, *Measurement of the inclusive differential cross section for* Z *bosons as a function of transverse momentum in* $\bar{p}p$ *collisions at* $\sqrt{s} = 1.8$ *TeV*, *Phys. Rev.* **D61** (2000) 032004 [arXiv:hep-ex/9907009].

[204] D0 collaboration, *Measurement of the shape of the boson transverse momentum distribution in* $p\bar{p} \to Z/\gamma^* \to e^+e^- + X$ *events produced at* $\sqrt{s}$=*1.96-TeV*, *Phys. Rev. Lett.* **100** (2008) 102002 [arXiv:0712.0803].

[205] D0 collaboration, *Measurement of the Normalized $Z/\gamma^* -> \mu^+\mu^-$ Transverse Momentum Distribution in $p\bar{p}$ Collisions at $\sqrt{s} = 1.96$ TeV*, *Phys. Lett.* **B693** (2010) 522 [arXiv:1006.0618].

[206] LHCb collaboration, *Measurement of the forward Z boson production cross-section in pp collisions at $\sqrt{s} = 7$ TeV*, *JHEP* **08** (2015) 039 [arXiv:1505.07024].

[207] LHCb collaboration, *Measurement of forward W and Z boson production in pp collisions at $\sqrt{s} = 8$ TeV*, *JHEP* **01** (2016) 155 [arXiv:1511.08039].

[208] CMS collaboration, *Measurement of the Rapidity and Transverse Momentum Distributions of Z Bosons in pp Collisions at $\sqrt{s} = 7$ TeV*, *Phys. Rev.* **D85** (2012) 032002 [arXiv:1110.4973].

[209] CMS collaboration, *Measurement of the transverse momentum spectra of weak vector bosons produced in proton-proton collisions at $\sqrt{s} = 8$ TeV*, *JHEP* **02** (2017) 096 [arXiv:1606.05864].

[210] ATLAS collaboration, *Measurement of the transverse momentum and ϕ^*_η distributions of Drell–Yan lepton pairs in proton–proton collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector*, *Eur. Phys. J.* **C76** (2016) 291 [arXiv:1512.02192].

[211] STAR collaboration, *Measurements of W and Z/γ^* cross sections and their ratios in $p + p$ collisions at RHIC*, [arXiv:2011.04708].

[212] PHENIX collaboration, *Measurements of $\mu\mu$ pairs from open heavy flavor and Drell-Yan in $p + p$ collisions at $\sqrt{s} = 200$ GeV*, *Phys. Rev.* **D99** (2019) 072003 [arXiv:1805.02448].

[213] S. Catani and M. Grazzini, *An NNLO subtraction formalism in hadron collisions and its application to Higgs boson production at the LHC*, *Phys. Rev. Lett.* **98** (2007) 222002 [arXiv:hep-ph/0703012].

[214] S. Catani, L. Cieri, G. Ferrera, D. de Florian and M. Grazzini, *Vector boson production at hadron colliders: a fully exclusive QCD calculation at NNLO*, *Phys. Rev. Lett.* **103** (2009) 082001 [arXiv:0903.2120].

[215] LHCb collaboration, *Measurement of the forward Z boson production cross-section in pp collisions at $\sqrt{s} = 13$ TeV*, *JHEP* **09** (2016) 136 [arXiv:1607.06495].

[216] ATLAS collaboration, *Measurement of the Z/γ^* boson transverse momentum distribution in pp collisions at $\sqrt{s}$ = 7 TeV with the ATLAS detector*, *JHEP* **09** (2014) 145 [arXiv:1406.3660].

[217] J. Pumplin, D. Stump and W. Tung, *Multivariate fitting and the error matrix in global analysis of data*, *Phys. Rev. D* **65** (2001) 014011 [arXiv:hep-ph/0008191].

[218] J. Dudek et al., *Physics Opportunities with the 12 GeV Upgrade at Jefferson Lab*, *Eur. Phys. J.* **A48** (2012) 187 [arXiv:1208.1244].

[219] D. Boer et al., *Gluons and the quark sea at high energies: Distributions, polarization, tomography*, [arXiv:1108.1713].

[220] T. Sjostrand, S. Mrenna and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **05** (2006) 026 [arXiv:hep-ph/0603175].