



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE
CORSO DI LAUREA MAGISTRALE IN FISICA
(LM-17)

**On the Impact of Data Inconsistencies on Parton
Distribution Determination**

Relatore:

Prof. Stefano Forte

Correlatore:

Dott. Andrea Barontini

Tesi di laurea di:

Giovanni De Crescenzo

matr. **982135**

a.a. 2022-2023

Contents

1	QCD and Parton Distribution Functions	7
1.1	Introduction to QCD	7
1.2	Renormalization of QCD	10
1.3	Factorization	13
1.4	Parton evolution	19
2	Error Propagation and Inverse Problems	21
2.1	Inverse problems formalism	21
2.2	Bayesian approach	23
2.3	Monte Carlo error propagation	24
3	Machine Learning and PDF Determination	27
3.1	Neural Networks	27
3.2	Neural Networks for PDF fitting: NNPDF	32
4	Methodology Validation: Closure Test	41
4.1	Closure test mechanics	41
4.2	Statistical test	42
5	Results	47
5.1	Consistent closure test results	47
5.2	Closure test revisited	51
5.3	Inconsistent closure test	57
5.3.1	Final remarks	78
A	Further results	83
A.1	DIS fits	84

A.2	DY fits	87
A.3	JETS fits	92

Introduction

The scope of this work is to gain further insight into the functioning of the Machine Learning algorithm developed by the NNPDF collaboration for the scope of Parton Distribution Function (PDF) determination. In particular, our aim is to understand the impact of inconsistent data on the methodology functioning.

The structure of the proton is encoded in the PDFs, which are not directly observable. In order to be able to compute theoretical predictions in QCD one needs then to infer their value starting from experimental measurements. In addition to the straightforward determination of the PDFs' values, it is essential to address the proper handling of uncertainties. This aspect forms the fundamental focus of our current research.

The framework of this thesis is the validation of the methodology known as closure test, which is employed by the NNPDF collaboration since 2012. In a closure test we choose an underlying true value in order to then generate several pseudo-data instances which are 'perfect', in the sense that they are free from internal inconsistencies since they are sampled from a totally known distribution. The next step is to corrupt this distribution in order to insert artificial inconsistencies in the generated pseudo-data: in principle this gives complete information on the response of the Neural Network to inconsistent data.

The first part of this thesis is actually devoted to the modification of the closure test setup: in particular, in order to quantify the performance of the NN in the closure test framework, it has been found that the previously adopted figure of merit was flawed by some problems. These issues have been partially solved with the redefinition of the figure of merit apt to deliver the judgement on the NN performance itself.

Having modified the previous way of evaluating the NN performance, we were able to extract the following results. We found that the NN behaves by learning the inconsistency only in extreme cases. Moreover, we found that the inconsistency is

propagated respecting a simple pattern: this means that the output of the inconsistently trained NN shows an inconsistent behaviour only on those processes which were made inconsistent in the input.

The thesis is outlined as follows: in Chapter 1, we introduce the fundamental concepts of Quantum Chromodynamics (QCD) and Parton Distribution Functions (PDFs). Our focus here is on exploring advanced topics within this field and elucidating their relevance to the present research.

Chapter 2 provides an overview of the methodology employed by NNPDF for error treatment. To do this, we briefly introduce the concept of inverse problems and detail the Bayesian approach used to address them. This will show the validity of the NNPDF approach in treating error.

Chapter 3 offers an introduction to Machine Learning and Neural Networks (NNs). We also present the essential technical details related to the NNPDF code, which forms the backbone of our study.

In Chapter 4, we delve into the methodology validation framework known as closure testing. This framework constitutes the main setting of the research proposed in this work.

Finally, Chapter 5 is devoted to the results of this thesis. The first part deals with the modification of the closure test formalism, and the explanation related to the need of such a change. The second part is then devoted to the presentation of the results concerning the inconsistent closure test.

Chapter 1

QCD and Parton Distribution Functions

This chapter is devoted to a brief introduction to the topic of Quantum ChromoDynamics (QCD) and Parton Distribution Functions (PDFs). The covered topics will regard technical aspects of the theory which are of fundamental importance in the context of this work. The basic notions of Quantum Field Theory are assumed to be familiar to the reader.

1.1 Introduction to QCD

QCD is the physical theory that describes strong interactions. The fundamental particles of the theory are the *quarks* which are the constituents of *hadrons*; these in turn include among many other particles the proton and the neutron just to cite a couple of well known examples.

Hadronic matter is divided into two families, *mesons* and *baryons*, which are respectively bound states of quark and anti-quark ($q\bar{q}$) and of three quarks (qqq). Quarks are spin 1/2 particles with fractionary charge whose basic properties are listed in the table below¹.

The properties listed in table (1.1) constitute what is known as the *naïve quark model*, which historically was the first one introduced. As it can be seen from the

¹It has to be said that the attribution of a mass to quarks is a tricky statement since quarks cannot be observed isolated.

Quark	Mass	Charge
Up (u)	$\sim 4 \text{ MeV}$	$+\frac{2}{3}$
Down (d)	$\sim 7 \text{ MeV}$	$-\frac{1}{3}$
Charm (c)	$\sim 1.5 \text{ GeV}$	$+\frac{2}{3}$
Strange (s)	$\sim 135 \text{ MeV}$	$-\frac{1}{3}$
Top (t)	$\sim 175 \text{ GeV}$	$+\frac{2}{3}$
Bottom (b)	$\sim 5 \text{ GeV}$	$-\frac{1}{3}$

Table 1.1: Quark properties

mass parameter, quarks can be divided into two families: *light* quarks which are u , d , s and *heavy* quarks which are the remaining three b , t , c . The light quarks obey an *approximate* symmetry under the group $SU(3)_f$ called *flavour symmetry*².

The *naïve* quark model briefly described until now poses a serious problem: some particles expected from the flavour symmetry needed to violate the Fermi-Dirac statistics. In particular in baryonic spectroscopy the resonance Δ^{++} was measured, which is a bound state of charge $2e$ and spin $3/2$. The problem with the state Δ^{++} is the fact that a state of three u quarks all with up spin would make it a totally symmetric state, violating Fermi-Dirac statistics.

In order to solve this problem a new quantum degree of freedom was introduced: the *colour number*, which is actually the fundamental property of QCD as will be seen later on. This new degree of freedom introduces another $SU(3)_c$ symmetry called symmetry of colour and the new index can take up three values usually called red, green and blue. Adding a new quantum number to the theory, the antisymmetry of the state Δ^{++} can be restored, regarding such a state as a totally anti-symmetric combination of the new degree of freedom:

$$|\Delta^{++}\rangle = \frac{1}{\sqrt{6}}\epsilon_{ijk}|u_i^\uparrow, u_j^\uparrow, u_k^\uparrow\rangle. \quad (1.1)$$

The introduction of this new degree of freedom immediately leads to another fundamental property of QCD which is *colour confinement*. There is resounding evidence that quarks cannot be observed isolated, thus the only observable hadronic states are bound states of quarks, which are exactly mesons and baryons.

²This division into light and heavy quarks is not totally rigorous. This split comes mainly from the fact that the mass corrections become important as the mass increases.

From a group theoretical point of view, colour confinement can be seen as a manifestation of the fact that $q\bar{q}$ (mesons) and qqq (baryons) all lie in the trivial representation of the colour symmetry group.

$$\text{Mesons : } 3_c \times \bar{3}_c = 8 \oplus 1, \quad (1.2)$$

$$\text{Baryons : } 3_c \times 3_c \times 3_c = 10 \oplus 8 \oplus 8 \oplus 1. \quad (1.3)$$

Lagrangian formulation of QCD

As any other Quantum Field Theory (QFT), also QCD is formalized through its Lagrangian formulation which will be shortly described here.

Each quark is represented as a Dirac spinor Ψ_f in the fundamental representation of $SU(3)_{fl}$, where f ranges over the possible flavours. Spinors also carry a colour index which refers to the colour symmetry $SU(3)_c$. It is upon quantization that QCD becomes a *non abelian gauge theory*: through the gauging of the symmetry of colour new particles are introduced. These are the *vector bosons* of the theory which are called *gluons*.

This is essentially what happens in Quantum Electro Dynamics (QED), with the only difference that here the symmetry group is $SU(3)_c$ which is non abelian and has higher dimension, yielding more than one gauge boson. Gluons are spin-1 particles described by 8 gauge fields A_a^μ living in the adjoint representation of $SU(3)_c$.

Similarly to the Lagrangian of QED, the Lagrangian of QCD reads as follows:

$$\mathcal{L} = \sum_{\text{flavours}} \bar{\Psi}_a (i\mathcal{D} - m_f)_{ab} \Psi_b - \frac{1}{4} F_{\alpha\beta}^a F_a^{\alpha\beta}. \quad (1.4)$$

$\mathcal{D} = \gamma_\mu D^\mu$ is the covariant derivative:

$$D^\mu = \partial^\mu + igT_a A_a^\mu \quad (1.5)$$

and $F_a^{\mu\nu}$ is the field strength tensor

$$F_a^{\mu\nu} = \partial^\mu A_a^\nu - \partial^\nu A_a^\mu - gf_{abc} A_b^\mu A_c^\nu. \quad (1.6)$$

In the equations above T_a are the generators of $SU(3)_c$ in the adjoint representation, g is the coupling constant and f_{abc} are the structure constants of the group. The following relation defines the structure constants relating them to the generators of the group:

$$[T^a, T^b] = if^{abc} T^c. \quad (1.7)$$

The Lagrangian is necessary in order to compute observables as a perturbative series in the coupling constant g . These computations though pose the problem of divergences in the series [18], cured by the known process of *renormalization*. Renormalization is a topic of fundamental importance in QFT and in the following section a brief discussion will be given.

1.2 Renormalization of QCD

Renormalization is closely related to the topic of PDFs for two main reasons: first of all it is through renormalization that one can understand the phenomenon of *asymptotic freedom*; second of all renormalization revolves around the formalism of the *renormalization group equations* and it will be shown that also PDFs need to undergo a certain renormalization procedure.

In QCD, as in typical, renormalization becomes necessary to address the issue of UV divergences which arise when calculating perturbative corrections to observables.



$$\propto \int \frac{d^4q}{q^2} \propto \Lambda^2$$

Figure 1.1: Loop diagram in vector boson self energy.

The coupling constant $\alpha := g^2/4\pi$ appearing in equation (1.4) is known as a *bare*, non observable quantity which holds no physical meaning. In order to have a sensible perturbation expansion of observable quantities such as cross sections, one needs to define a *physical* coupling constant α_s . There are various possible approaches to renormalization: just to keep the discussion general it can simply be said that α_s needs the choice of some renormalization point μ which will then become another scale parameter of the theory.

To briefly introduce the topic, consider a dimensionless observable R involving a scale Q computable in perturbation theory. For the forthcoming discussion until the end of the chapter the masses of the fundamental particles of the theory will be considered ~ 0 , thus they do not constitute a fundamental scale of the theory itself.

From dimensional analysis R can only be a function of this form:

$$R = R(\mu, Q, \alpha_s(\mu)) = R(Q^2/\mu^2, \alpha_s). \quad (1.8)$$

R is an observable quantity thus it cannot depend upon the choice of the renormalization point μ ; this independence defines then the *renormalization group equation* for observables:

$$\mu^2 \frac{d}{d\mu^2} R(Q^2/\mu^2, \alpha_s) = \left[\mu^2 \frac{\partial}{\partial \mu^2} + \mu^2 \frac{\partial \alpha_s}{\partial \mu^2} \frac{\partial}{\partial \alpha_s} \right] R = 0. \quad (1.9)$$

Introducing some variables (1.9) can be rewritten in a simpler form. Define

$$t = \ln\left(\frac{Q^2}{\mu^2}\right), \quad \beta(\alpha_s) = \mu^2 \frac{\partial \alpha_s}{\partial \mu^2} \quad (1.10)$$

which in turn yield for (1.9):

$$\left[-\frac{\partial}{\partial t} + \beta(\alpha_s) \frac{\partial}{\partial \alpha_s} \right] R(e^t, \alpha_s) = 0. \quad (1.11)$$

This differential equation can be implicitly solved by defining a new function, the *running coupling* $\alpha_s(Q^2)$:

$$t = \int_{\alpha_s}^{\alpha_s(Q^2)} \frac{dx}{\beta(x)}, \quad \alpha_s(Q^2 = \mu^2) = \alpha_s. \quad (1.12)$$

Upon differentiation of (1.12) the following is obtained:

$$\frac{\partial \alpha_s(Q^2)}{\partial t} = Q^2 \frac{\partial \alpha_s(Q^2)}{\partial Q^2} = \beta(\alpha_s), \quad (1.13)$$

which is a renormalization group equation for the running coupling $\alpha_s(Q^2)$. β is a quantity which has a perturbative expansion in α_s :

$$\beta(\alpha_s) = \alpha_s^2 \left(-\beta_0 + \sum_k \beta_k \alpha_s^k \right). \quad (1.14)$$

The running coupling equation can then be perturbatively solved: its solution gives information regarding the high and low energy limits of the coupling constant. Without addressing the problem of relating the coefficients β_k to the characteristics of the theory, and keeping only the first order of eq. (1.13), we get to

$$Q^2 \frac{d\alpha_s(Q^2)}{dQ^2} = -\beta_0 \alpha_s^2, \quad (1.15)$$

which can be solved remembering the boundary condition for α_s in Eq. (1.12).

$$\int_{\alpha(\mu^2)}^{\alpha(Q^2)} \frac{d\alpha(Q^2)}{-\beta_0\alpha(Q^2)} = \int_{\mu^2}^{Q^2} \frac{dQ^2}{Q^2},$$

$$\frac{1}{\beta_0} \left(\frac{1}{\alpha^2(Q^2)} - \frac{1}{\alpha^2(\mu^2)} \right) = \ln\left(\frac{Q^2}{\mu^2}\right), \quad (1.16)$$

$$\frac{1}{\alpha^2(Q^2)} = \frac{\beta_0\alpha^2(\mu^2) \ln\left(\frac{Q^2}{\mu^2}\right) + 1}{\alpha^2(\mu^2)}.$$

In the end this yields:

$$\alpha^2(Q^2) = \frac{\alpha^2(\mu^2)}{\beta_0\alpha^2(\mu^2) \ln\left(\frac{Q^2}{\mu^2}\right) + 1}. \quad (1.17)$$

Equation (1.17) gives the trend of the running coupling α_s for increasing Q^2 which determines whether the theory is *asymptotically free* or not. In QCD

$$\beta_0 = \frac{33 - 2n_f}{12\pi}, \quad (1.18)$$

where n_f is the number of active light flavours. In particular in QCD $n_f \leq 16 \implies \beta_0 > 0$, which in turn implies that α_s becomes smaller as Q^2 increases. This property is known as asymptotic freedom. Clearly for decreasing Q^2 there is a scale Λ at which the perturbative expansion breaks, which in QCD has been found from experiments to be $\Lambda \sim 200 \text{ MeV}$ ³. Given the asymptotic freedom phenomenon and the existence of a scale at which the perturbative expansion breaks, we can split QCD in a *perturbative* sector and a *non perturbative* one.

The role of the sign of β_0 is of fundamental importance, since if $\beta_0 < 0$ all the discussion above would work in the opposite way.

Renormalization is necessary in order to understand if processes at high energies can be treated perturbatively, which is the case for QCD. The complication arises though since every hadronic process includes bound states of quarks due to colour confinement: this phenomenon is a manifestation of the low-momentum sector of QCD, thus it cannot be treated perturbatively. This problem is overcome thanks to the factorization theorem and the introduction of Parton Distribution Functions, topics which will be briefly introduced in the coming chapters.

³ $\Lambda \sim 200 \text{ MeV}$ is an extreme limit for the fail of the perturbative approach; it is safe to say that it actually starts failing at much higher energies, $\sim 1 \text{ GeV}$.

1.3 Factorization

It is thanks to factorization that one is able to actually compute the values of the PDFs. Factorization roughly states that observables can be computed *factorizing* effects of perturbative QCD and non-perturbative QCD.

Hadronic processes can be very generally distinguished between processes with one hadron in the initial state or more than one hadron (generally two). We start by giving the description of a particular single hadron process, a Deep Inelastic Scattering event. From the study of this process we will be able to introduce both PDFs and the factorization theorem.

Deep Inelastic Scattering

Deep Inelastic lepton-hadron scattering is the simplest process that gives useful insight in the topic of factorization and PDFs. The scattering becomes *inelastic* if the momentum transfer Q is high enough to disintegrate the target hadron. Let us consider a specific case: a charged lepton (an electron) of momentum k scattering off a target proton of momentum P . The basic diagram of the process is:

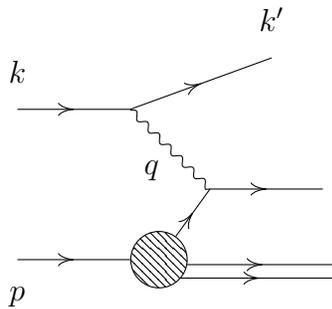


Figure 1.2: Deep Inelastic Scattering diagram.

and the kinematic variables are usually renamed as follows:

- $Q^2 = -q^2$
- $M^2 = p^2$
- $\nu = p \cdot q$
- $x = \frac{q^2}{2\nu}$

- $y = \frac{q \cdot p}{k \cdot p}$

The differential cross section of the process shown in figure (1.2) can be written as follows:

$$\frac{d^2\sigma}{dx dy} = \frac{8\pi\alpha^2 ME}{Q^4} \left[\left(\frac{1 + (1-y)^2}{2} \right) 2xF_1^{em} + (1-y)(F_2^{em} - 2xF_1^{em} - (M/2E)xyF_2^{em}) \right]. \quad (1.19)$$

F_i^{em} are the *electromagnetic structure functions*, which in some way ‘parametrize’ the structure of the target as seen from the incoming particle. The presence of such functions starts giving an idea for the successive parton model of the proton. Before formulating the *naïve* parton model, the so called *Bjorken scaling* has to be postulated.

Experimental data roughly confirm that structure functions obey an approximate scaling law, that is they depend only on the variable x when $Q^2, \nu \rightarrow \infty$ with x fixed, that is:

$$F(x, Q^2) \xrightarrow{Q^2, \nu \rightarrow \infty} F(x). \quad (1.20)$$

Below a collection of data supporting this fact.

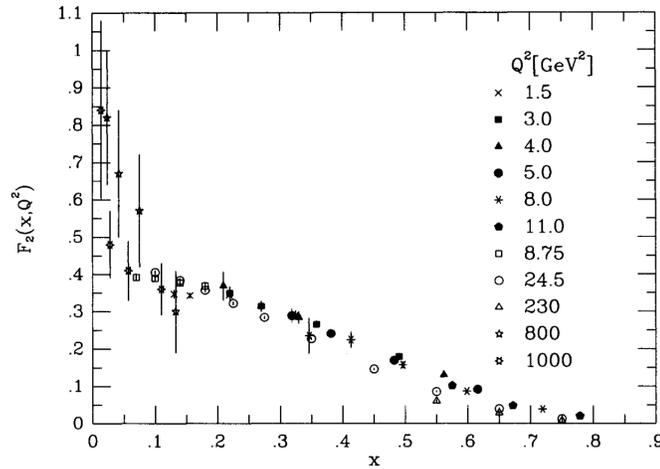


Figure 1.3: Image taken from [15]. Measurements for the F_2 structure function from various collaborations. As it can be seen to a good approximation the values lie on a universal curve even for different Q values

Bjorken scaling implies that leptons scatter off pointlike constituents, since otherwise the structure functions would have a dependence on Q through the ratio Q/Q_0

where $1/Q_0$ sets some size scale for the struck proton. In the limit of massless proton $p \sim (P, 0, 0, P)$ with $P \gg M$, equation (1.19) can be rewritten as:

$$\frac{d^2\sigma}{dx dy} = \frac{4\pi\alpha^2}{Q^4} \left[1 + (1-y)^2 F_1 + \frac{1-y}{x} (F_2 - 2xF_1) \right]. \quad (1.21)$$

The *naïve* parton model can be easily introduced having written down (1.21). Take into consideration the unpolarized cross section $e^- + q \rightarrow e^- + q$; its squared amplitude takes the form:

$$\overline{\sum} |M|^2 = 2e_q^2 e^4 \frac{\hat{s}^2 + \hat{u}^2}{\hat{t}^2}, \quad (1.22)$$

where the $\hat{}$ stands for the quark level Mandelstam variables and the $\overline{\sum}$ is the sum over final and average over initial states. Relating now this simple $2 \rightarrow 2$ scattering to (1.22), suppose the quark of $e^- + q \rightarrow e^- + q$ carries a fraction of the proton momentum, that is $p_q = \xi p$ with $0 < \xi < 1$. From (1.21) the totally differential cross section can be computed, which yields:

$$\frac{d^2\hat{\sigma}}{dx dy} = \frac{4\pi\alpha^2}{Q^4} \left[1 + (1-y)^2 \right] \frac{1}{2} e_q^2 \delta(x - \xi). \quad (1.23)$$

Reinserting the notation for structure functions in this context, the *quark level* structure functions can be introduced. Comparing with eq.(1.21) these would be:

$$\hat{F}_2(x) = 2x\hat{F}_1(x) = xe_q^2\delta(x - \xi). \quad (1.24)$$

This simple example suggests that the structure function F_2 ‘probes’ a quark with momentum fraction ξ . From experimental data it can be clearly seen that in the case of DIS the structure function is a broader distribution rather than a δ , which can be interpreted as:

- the quarks constituents carry a range of momentum fractions,
- $F_2(x)$ for the proton has to be *weighted* by the probability of finding a quark with ξ momentum fraction.

This gives rise to the *naïve parton model*. The probability of finding a quark with momentum fraction ξ is define as $q(\xi) d\xi$, which then yields:

$$\begin{aligned} F_2 = 2xF_1 &= \sum_{q,\bar{q}} \int_0^1 d\xi q(\xi) \hat{F}_2(\xi) \\ &= \sum_{q,\bar{q}} e_q^2 x q(x). \end{aligned} \quad (1.25)$$

Up until this point nothing has been said regarding *factorization*. What can be seen is that (1.21) is actually factorized into a short distance component (the structure functions) and into a long distance one (the rest). The validity of this statement can be seen to hold to any order in perturbation theory, but its proof is outside the scope of the present work. For our purposes it is only important to state that any DIS cross section can be written down as:

$$\sigma^{lh} = \sum_i \sum_f \int dx_i \int d\Phi_f q_i(x_i, Q_F^2) \frac{d\sigma^{lp \rightarrow f}(x_i, \Phi_f, Q_F^2)}{dx_i dQ_F^2}. \quad (1.26)$$

Collinear Next to Leading Order DIS

In this subsection we will treat the Next to Leading Order (NLO) collinear corrections to the DIS process described above. This will both prove the breaking of Bjorken scaling at higher order in perturbation theory and allow us to introduce the evolution equations for the PDFs. To incorporate the collinear NLO corrections, we must begin by introducing additional particles into the interaction vertex. It is at this point that perturbative QCD begins playing a role in the cross-section calculation.

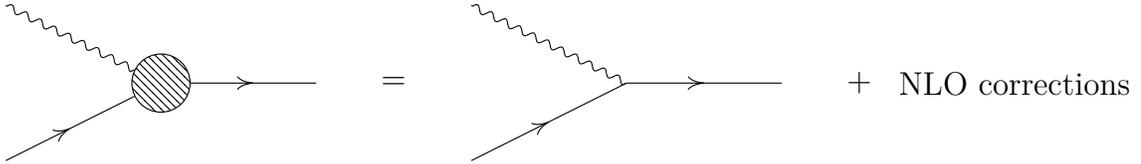


Figure 1.4: QCD vertex

The next more complicated process is a gluon emission by the quark.



Figure 1.5: NLO collinear singularities arising from real emission of gluons

These vertex corrections give rise to various contributions to the square matrix element; in figure 1.6 we show the only divergent one. Without getting into the technical aspect of the calculation, the phase space for this contribution can be written

down as:

$$d\Phi_2 = \frac{1}{4\pi^2} \int d^4k \delta^+((p-k)^2) \delta^+((k+q)^2). \quad (1.27)$$

Introducing a vector n^μ and k_T^μ , k and d^4k can be rewritten as:

$$k^\mu = \xi p^\mu + \frac{k_T^2 - |k^2|}{2\xi} n^\mu + k_T^\mu, \quad (1.28)$$

$$d^4k = \frac{d\xi}{2\xi} dk^2 d^2k_T, \quad (1.29)$$

which in turn yields for the phase space:

$$d\Phi_2 = \frac{1}{16\nu\pi^2} \int d\xi dk^2 d^2k_T d\theta \delta(k_T^2 - (1-\xi)|k^2|) \times \\ \times \delta\left(\xi - x - \frac{|k^2| + 2q_T \cdot k_T}{2\nu}\right), \quad (1.30)$$

where $\theta \in (0, \pi)$. Knowing the phase space, the square amplitude contribution can be

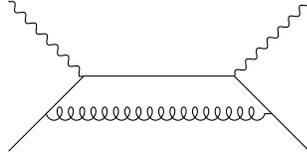


Figure 1.6: Divergent contribution from real gluon emission

calculated. Skipping the technicalities we find that:

$$\frac{1}{4\pi} n^\alpha n^\beta \overline{\sum} |\mathcal{M}_{\alpha\beta}|^2 = \frac{8e_q^2 \alpha_s}{|k^2|} \xi P(\xi), \quad (1.31)$$

where $|\mathcal{M}_{\alpha\beta}|^2$ is the matrix element of the scattering process, which is then multiplied by n_α and n_β to project out the contribution to \hat{F}_2 . $P(\xi)$ is known as *parton splitting function* whose form is characteristic of the qqg vertex:

$$P(\xi) = \frac{4}{3} \frac{1 + \xi^2}{1 - \xi}. \quad (1.32)$$

The parton level structure function can be extracted from (1.31) and (1.30):

$$\hat{F}_2 = e_q^2 \frac{\alpha_s}{2\pi^2} \int_0^{2\nu} \frac{d|k^2|}{|k^2|} \int_{\xi^-}^{\xi^+} d\xi \frac{\xi P(\xi)}{\sqrt{(\xi^+ - \xi)(\xi - \xi^-)}}, \quad (1.33)$$

where $\xi^{+/-}$ are:

$$\xi^{+/-} = x + z - 2xz \pm \sqrt{4x(1-x)z(1-z)}, \quad (1.34)$$

where $z = |k^2|/2\nu$.

It can be seen that (1.33) is actually divergent for $k \rightarrow 0$. Regularising the divergence with a small cut off ϵ the parton level structure function takes the following form:

$$\hat{F}_2(x, Q^2) = e_q^2 x \left[\delta(1-x) + \frac{\alpha_s}{2\pi} \left(P(x) \ln\left(\frac{Q^2}{\epsilon} + C(x)\right) \right) \right]. \quad (1.35)$$

This shows that beyond leading order Bjorken scaling is broken by logarithms of Q in the structure function. This divergence in the structure function cannot be treated as the other divergences: here come into play the PDFs. The proton level structure function has to be regarded as a convolution between (1.35) and a *bare* parton distribution function $q_0(\xi)$ which absorbs the divergence of \hat{F}_2 . This yields:

$$F_2(x, Q^2) = x \sum_{q, \bar{q}} e_q^2 \left[q_0(x) + \frac{\alpha_s}{2\pi} \int_x^1 \frac{d\xi}{\xi} q_0(\xi) + \left\{ P\left(\frac{x}{\xi}\right) \ln \frac{Q^2}{\epsilon} + C\left(\frac{x}{\xi}\right) \right\} + \dots \right], \quad (1.36)$$

and similarly to coupling constants the *physical* parton distribution is defined at a factorization scale μ^2 :

$$q(x, \mu^2) = q_0(x) + \frac{\alpha_s}{2\pi} \int_x^1 \frac{d\xi}{\xi} q_0(\xi) \left\{ P\left(\frac{x}{\xi}\right) \ln \frac{\mu^2}{\epsilon} + C\left(\frac{x}{\xi}\right) \right\} + \dots \quad (1.37)$$

These functions cannot be calculated in perturbation theory since they receive contributions from the low-momentum sector of the strong interaction. They need then to be *fitted* to experimental data of observables (such as the structure functions above). Given the introduction of a new scale μ , also PDFs obey some kind of RGE: these are known as D'Altarelli, Parisi (DGLAP) evolution equations, which will be the topic of the next section. Before getting into said topic, let us briefly mention how factorization is treated for the hadronic processes.

Hadronic processes

Everything said up until now concerns only DIS processes, which involve only one hadron in the initial states. Thanks to the factorization theorem also a scattering

between two hadrons can be factorized in long-distance and short-distance behaviour. The scattering process of two hadrons can be generally written down as follows:

$$h_1(p_1) + h_2(p_2) \longrightarrow W(Q^2) + X, \quad (1.38)$$

where W and X respectively refer to an exclusive and inclusive part of the final state.

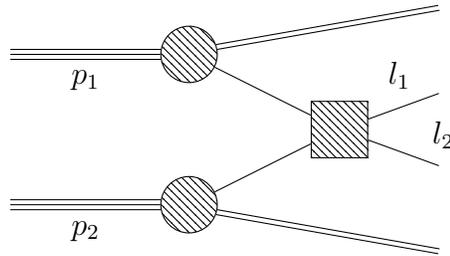


Figure 1.7: Drell-Yan process with lepton pair production.

Being there two hadrons in the initial state the factorization theorem will yield a more complicated formula for the factorized cross section, which involves two sets of PDFs. A general hadronic cross section takes then the following form:

$$\sigma(p_1, p_2) = \sum_{i,j} \int dx_1 dx_2 q_i(x_1, \mu^2) q_j(x_2, \mu^2) \hat{\sigma}_{q_i q_j \rightarrow l^+ l^-}, \quad (1.39)$$

where $\hat{\sigma}$ is the parton level cross section and $p_i = \xi P_i$ as in the DIS case. An example of a hadronic process can be lepton pair production in what is called to be a *Drell-Yan* process, which is shown in the diagram (1.7)

1.4 Parton evolution

The full form of evolution equations can be justified only through a Wilson Product Expansion. For the scope of this thesis it is only useful to actually write down the DGLAP equations, which are a set of coupled partial differential equations:

$$t \frac{\partial}{\partial t} \begin{bmatrix} q_i(x, t) \\ g(x, t) \end{bmatrix} = \frac{\alpha_s(t)}{2\pi} \sum_{q_j, \bar{q}_j} \int_x^1 \frac{d\xi}{\xi} \times \begin{bmatrix} P_{q_i, q_j}(\frac{x}{\xi}, \alpha_s(t)) & P_{q_i, g}(\frac{x}{\xi}, \alpha_s(t)) \\ P_{g, q_j}(\frac{x}{\xi}, \alpha_s(t)) & P_{g, g}(\frac{x}{\xi}, \alpha_s(t)) \end{bmatrix} \begin{bmatrix} q_i(x, t) \\ g(x, t) \end{bmatrix}, \quad (1.40)$$

where each *splitting function* $P_{q,g}$ is calculable as a power series in the coupling. The system of coupled equations of (1.40) can be partially de-coupled introducing a new basis called *evolution* basis. This de-coupling is possible thanks to a few properties of the splitting functions coming from physical laws which have to be respected [15].

The evolution basis is defined as follows:

$$\begin{aligned}
V_i &= q_i^- \\
T_3 &= u^+ - d^+ \\
T_8 &= u^+ + d^+ - 2s^+ \\
T_{15} &= u^+ + d^+ + s^+ - 3c^+ \\
T_{24} &= u^+ + d^+ + s^+ + c^+ - 4b^+ \\
T_{35} &= u^+ + d^+ + s^+ + c^+ + b^+ - 5t^+
\end{aligned} \tag{1.41}$$

where $q_i^\pm = q_i \pm \bar{q}_i$. The only one remaining contribution is given by the *singlet* distribution defined as:

$$\Sigma = \sum_i q_i^+, \tag{1.42}$$

which is coupled to the gluon evolution. Knowing then the evolution for the 6 V_i , the 5 T_k and the singlet Σ one can compute the evolution of the 12 individual PDFs, which become 13 when we introduce the gluon one.

Chapter 2

Error Propagation and Inverse Problems

The previous chapter was devoted to the introduction of the PDFs, whose numerical determination is at the core of the present work. As already said in the introduction a point of fundamental importance is the propagation of error from the experimental data to the fitted PDFs themselves.

It becomes then essential to have a good methodology for propagating error: following the steps presented in [13] we are going to show the validity of the method chosen by NNPfD for propagating error.

In order to do so we are going to lay down some basics regarding a Bayesian approach to inverse problems [19]. Roughly speaking an inverse problem corresponds to a situation in which we have a measurement of some quantity Y which depends on another quantity X which we want to know. It is clear then that PDF determination falls under this category of problems, being Y the experimental data and X the PDFs themselves.

2.1 Inverse problems formalism

Call X the *input space* and Y the *output space*. In the case of the present work these spaces respectively denote the space where PDFs live and the space of observables. Notice that both X and Y are sets of *functions*: in order to keep the discussion rigorous these spaces should be regarded as Banach spaces, even if in the end just a finite approximation will be kept.

These spaces are connected by a map, which we shall call the *observable map* M :

$$M : X \rightarrow Y. \quad (2.1)$$

This map represents in general any physical observable depending on objects living in X . For the case of PDFs an example can be the structure functions of hadronic cross sections at NLO already described in chapter (1), such as:

$$M : (q_i(x, Q^2), \bar{q}_i(x, Q^2), g(x, Q^2)) \mapsto F_2(x, Q^2). \quad (2.2)$$

In general we denote by M the function which maps the PDFs to all the possible physical observables involving them. In order to make notation lighter a general element belonging to X will be denoted as u , thus:

$$M : u \mapsto M(u). \quad (2.3)$$

Actual experiments do not have access to the whole image-functions living in Y . This is common to any experimental setup, since experiments necessarily consist in a finite amount of measurements. To formalize this, a second map \mathcal{O} can be introduced:

$$\mathcal{O} : Y \rightarrow \mathbb{R}^{N_{\text{data}}}, \quad (2.4)$$

where in general N_{data} is equal to the number of actual measurements taken into consideration. Finally the composition between the two maps formalizes the actual *measurement operation* of an experiment:

$$O := \mathcal{O} \circ M. \quad (2.5)$$

Furthermore every measurement is affected by random noise, which means that experimental central values can be written as:

$$y_0 = O(u) + \eta \quad (2.6)$$

where η is a Random Variable (R.V.) distributed according to some probability density function ρ . In the context of this work we will always deal with a multi-variate normal:

$$\rho = \mathcal{N}(0, C_{\text{exp}}), \quad (2.7)$$

where C_{exp} is the experimental covariance matrix.

Starting from a set of experimental central values y_0 the goal is to find a sensible value for u knowing the map O : this is roughly the definition of inverse problem. The main complication in this kind of inverse problem is two-fold:

- The complexity of the forward map O .
- The presence of noise in the measurement.

The presence of noise leads to the need for propagation of error in the determination of PDFs. In order to show the validity of the method employed by NNPf we are going to treat this inverse problem following a Bayesian approach.

2.2 Bayesian approach

In order to simplify the discussion, $X = \mathbb{R}^{N_{\text{model}}}$ where N_{model} is the number of parameters defining the model, which in the case of PDFs consists in the parametrization of the functions, topic which will be discussed in later chapters regarding the specifics of the code implementation.

In a Bayesian approach the goal is to define a *posterior* probability measure $\mu_X(u|y)$ of the model u given a noisy observation of some data y as in (2.6). The crucial passage in order to define such a measure of probability is to make use of Bayes' theorem, which in its most general form can be written down as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \propto P(B|A)P(A), \quad (2.8)$$

$$\mu_X(u|y) \propto \mu_Y(y|u)\mu_X^0(u). \quad (2.9)$$

The *unconditioned* probability $P(A)$ is known as the *prior probability*. In order to translate Bayes' theorem to the case of interest, first of all we need to encode some kind of prior knowledge of the model in what is known as the *prior measure* $\mu_0(u)$. In second place the conditional measure $\mu_Y(y|u)$ can be found simply by looking at the definition of y_0 given in (2.6). In the end Bayes' theorem yields:

$$\mu_X(u|y) \propto \mu_X^0(u)\rho(O(u) - y). \quad (2.10)$$

Let us now specialize the discussion to the case which will be taken into consideration: consider O to be a *linear map*, thus a matrix, and consider also the prior μ_X^0 to be gaussian:

$$\mu_X^0(u) = \exp\left(-\frac{1}{2}(u - u_0)_i C'^{-1}_{ij} (u - u_0)_j\right), \quad (2.11)$$

where the model $u \in \mathbb{R}^{N_{\text{model}}}$ and where C' denotes some covariance related to the model prior. Suppose also that the probability density function of the noise ρ is

Gaussian:

$$\rho(\eta) = \exp\left(\eta_i C_{ij}^{-1} \eta_j\right), \quad (2.12)$$

where C would stand for the experimental covariance matrix in the particular case of PDF fitting. Equation (2.10) becomes then:

$$\mu(u|y) \propto \exp\left(|u - u_0|_{C'} + |y - G(u)|_C\right) = \exp(S(u)). \quad (2.13)$$

Considering the case of *uninformative prior*, we get to the analytical expression for the covariance and mean of the posterior distribution in equation (2.13):

$$C_X = \left(O^T C_Y^{-1} G\right), \quad (2.14)$$

$$\bar{u} = C_X O^T C_Y^{-1} y_0. \quad (2.15)$$

It is under these assumptions that we are going to show the equivalence of the NNPDF approach for propagating error.

2.3 Monte Carlo error propagation

In this section we want to give more details to the statistical properties of the NNPDF approach to the inverse problem. In particular the core of the discussion revolves around the treatment of experimental error and its propagation to the fitted PDFs. Broadly speaking the NNPDF framework adopts what is generally known as a *Monte Carlo* (MC) approach for error propagation: broadly speaking a Monte Carlo approach delivers error propagation directly by generating a *distribution* of the final object which we are fitting. By looking at the properties of the sample of this distribution one can then estimate the error. The MC approach is opposed to the Hessian approach for error propagation which has been thoroughly described in [12].

The starting input of any fitting procedure are the central values of an experiment y_0 and the correlated noise defined by the measure of probability ρ . In order to use this information to both infer the value of the PDFs and their error we need to generate some distribution of PDFs as said in the introduction. First of all we generate several pseudo-data replicas as written below:

$$\mu^{(k)} = y_0 + \epsilon^{(k)}, \quad (2.16)$$

where the additive noise ϵ is taken from the same probability distribution as the experimental noise, i.e. $\mathcal{N}(0, C_{\text{exp}})$.

After generating this set of pseudo-data a *model replica* is fitted to each *data replica* by minimizing the χ_k^2 between predictions and the replica values:

$$u_*^{(k)} = \operatorname{argmin}_{u \in X} \chi_k^2. \quad (2.17)$$

The replica chi square is defined as:

$$\chi_k^2 := (\mu^{(k)} - O(u))^T C_Y^{-1} (\mu^{(k)} - O(u)). \quad (2.18)$$

First of all we need to properly address the definition of the map O in the context of PDFs. We need to perform some *linearization* operation in order to actually get to the result of this sections.

As explained before consider the model space X to be finite dimensional, so:

$$X = \mathbb{R}^{N_{\text{model}}}, \quad (2.19)$$

which means discretizing the PDFs on a grid of points:

$$u \mapsto u(x_i) \quad i = 1 \dots N_{\text{model}}. \quad (2.20)$$

Consider linear observables such as structure functions, which at LO can then be simply considered linear maps F of the parametrization of the PDFs:¹

$$y_i = \sum_{j=1}^{N_{\text{model}}} F_{ij} u_j. \quad (2.21)$$

In order to get analytical results for error propagation, the second approximation needed is hidden in the NN parametrization of the u_j values. The value of the PDFs at each point can be thought of as a function of the set of weights $\vec{\theta}$ of the NN:

$$u_j = u(x_j) = u(x_j, \vec{\theta}). \quad (2.22)$$

Then again if the weights have a peaked enough distribution around some central value $\bar{\theta}$ the following approximation is valid:

$$y_i = \sum_{j=1}^{N_{\text{model}}} F_{ij} \left[u_j(\bar{\theta}) + \sum_k \frac{\partial u}{\partial \theta_k} (\theta - \bar{\theta})_k \right]. \quad (2.23)$$

¹As in [13] we are not going to explicitly talk about hadronic observables, which are linear in the PDFs; also for that case though one can simply expand one more time the observable map around the most peaked value linearizing that map too.

In order to retrieve the previously adopted notation, call O the linearized map:

$$O := \sum_j F_{ij} \frac{\partial u}{\partial \theta_k} \quad (2.24)$$

The final assumption needed to get analytical results for the NNPDF propagation is the requirement that the matrix F has linearly independent rows.

We can now get to the aforementioned equivalence between this approach and the Bayesian posterior. By explicit minimization of (2.18) we get:

$$u_*^{(k)} = (O^T C_Y^{-1} O)^{-1} (O^T C_Y^{-1} y_0 + O^T C_Y^{-1} \epsilon^{(k)}), \quad (2.25)$$

which is the explicit form of the model in term of both the noise and the central values. Since $u_*^{(k)}$ depends *linearly* on the gaussian R.V. ϵ , also u_* is gaussianly distributed. It is easy to compute the mean and covariance of this gaussian, which read:

$$C_X = \left(O^T C_Y^{-1} G \right), \quad (2.26)$$

$$\bar{u} = C_X O^T C_Y^{-1} y_0. \quad (2.27)$$

These are exactly the same defining moments as the posterior probability distribution given by the Bayesian approach in equation (2.15).

Chapter 3

Machine Learning and PDF Determination

This chapter is devoted to a brief introduction to Machine Learning, specifically focusing on Neural Networks, which are at the core of the methodology for PDF determination used in the context of this thesis. In this brief introduction first of all we are going to give the general idea of functioning of Neural Networks in order to then specialize the discussion to the code used in the context of this work.

3.1 Neural Networks

Computer simulations have gained importance in the field of physics given their capability to solve numerically intricate problems: some examples can be the solution of differential equations to predict the evolution of some dynamical system or also problems in the area of statistical mechanics. Machine Learning and more specifically Neural Networks deviate from the normal way of programming as will be briefly explained in the subsequent section. Clearly this really brief introduction does not give justice to this field which is extremely wide and complex: the only goal here is to give a rough explanation of the topic in order to make the reader able to follow the rest of this work.

Standard programming

A *classical* program could be summarized as a set of *rules* according to which a set of *inputs* is transformed to give a certain *output*. A really simple *flowchart* of this structure is depicted in figure (3.1).



Figure 3.1: Simple classical program flowchart

A practical example could be the resolution of the problem of the time evolution of a dynamical system: knowing the physical rules describing the motion of the system and given the initial conditions, the system can be simulated yielding its evolution through time.

Machine Learning

Machine Learning works following a different philosophy. In the context of physics, Machine Learning can be employed in situations in which the rules of Nature are not known in order to try to understand them. Roughly speaking, when programming a Machine Learning *tool* what is actually implemented by the programmer is an *architecture* which contains a variety of possible rules. This set of rules is explored, and a particular element is then chosen during the so-called *training phase*. The best set of rules is chosen according to some specific metric which is dependent on the specific case. Just to give an over-used example consider the problem of writing a program which classifies a set of images representing cats and dogs, labeling each image as C if it represents a cat and as D if it represents a dog. Clearly we do not know the function which associates a random image of 28x28 pixels to the correct class C or D. A Machine Learning program in this case can be thought of as an ensemble comprising *all*¹ the possible functions which divide *all* possible images into the two classes: then during the training phase a set of images already labeled as C or D are used in order to choose which function best approximates the already given labeling. This best approximation is performed by minimizing what is known as the

¹The topic of the power of a Machine Learning tool is clearly complex, and by ‘any’ we mean that the set of possibilities is very wide.

loss function, which can be defined as a metric which quantifies the distance between the predictions of the Machine Learning tool and expected labels.

Neural Networks

An important subclass of Machine Learning are Neural Networks. Neural Networks can be thought of as functions parametrized by a set of parameters:

$$\text{NN} : \mathbb{R}^{N_{\text{input}}} \rightarrow \mathbb{R}^{N_{\text{output}}}, \quad (3.1)$$

$$f(x, \theta) : x \mapsto y. \quad (3.2)$$

The basic unit of a Neural Network is what is called a neuron. A neuron is a simple operator whose action on the input is defined by a set of weights and by an *activation function*. Given a multi-dimensional numerical input x , the neuron's output is determined by:

$$\text{Neuron} : x \mapsto y = f_{\text{activation}}(W^T \cdot x), \quad (3.3)$$

where W is the set of parameters called weights and $f_{\text{activation}}$ is the activation function. Usually f is chosen from a set of commonly used ones such as the sigmoid. The output of a Neural Network is determined by a successive composition of the Neuron's functions: the number of neurons and of connections between them is what completely defines the action of the NN. Below is reported a simple example. In (3.2) the action

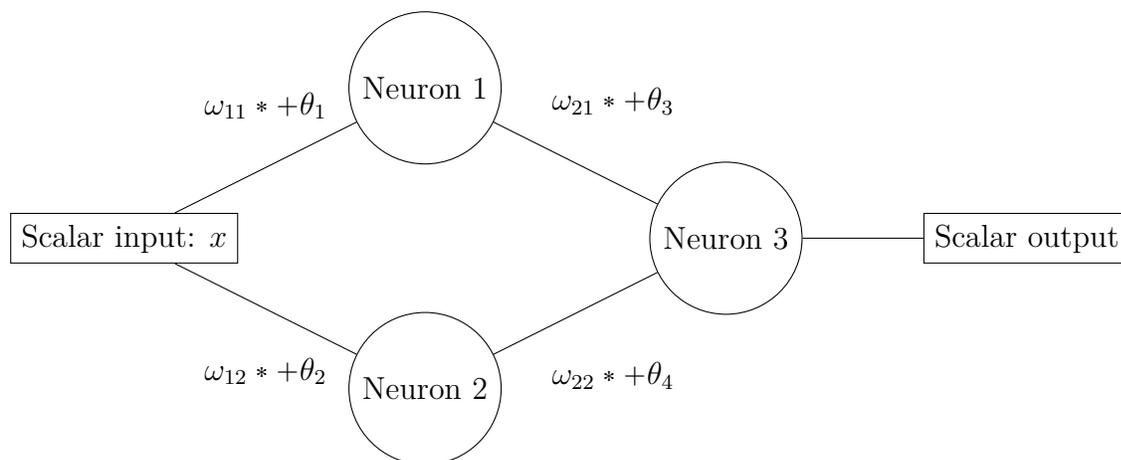


Figure 3.2: simple NN 1-2-1. The symbol $*$ represents the output coming from the start of the connection

of the NN on the input is represented graphically. The scalar input is modified by neuron 1 and neuron 2; successively the 2 intermediate outputs are combined and used as inputs of neuron 3, which then determines the output of the NN. Given a certain activation function for each neuron called $g(x)$, the output of the NN would be a successive composition of this function:

$$y = g(\omega_{21}g(\omega_{11}x + \theta_1) + \omega_{22}g(\omega_{12}x + \theta_2) + \theta_3 + \theta_4), \quad (3.4)$$

where ω_{ij} and θ_i are the free parameters determining the action of each neuron. It is clear from (3.4) that the parameters are what define the output of the Neural Network and these are the parameters that are modified during the training phase of the algorithm. The example in (3.2) is really simple, but the number of parameters and the complexity of the final function can be *arbitrarily*² enhanced by making the network more complex.

Just to give more terminology of NN the example of (3.2) is said to have one *layer* of Neurons of *width* 2. The structure of the NN can be made more complex by adding neurons and disposing them in different layers. As the number of neuron increases the analytical form of the NN ‘map’ becomes too intricate to even write down. The potential of a NN to represent any kind of function is what goes under the name of ‘universal approximation theorem’ which roughly states that any function can be approximated arbitrarily well by a complex enough NN.

Training phase: comparison between classical approach and NN

Consider the following inverse problem, which has already been introduced in the previous chapter: given a set of data subject to noise the goal is to infer the underlying function f which the data indirectly measure:

$$D_i = O_i(f(x)) + \eta, \quad (3.5)$$

where D_i indicate the measurements, η is the sample of a Random Variable representing noise and O is a known functional which maps f to the data:

$$O_i : f \mapsto D_i. \quad (3.6)$$

Here we want to take a slight detour including in the discussion also the classical way of approaching the problem of inferring a function given some noisy measurement.

²As before there are limits to the arbitrariness in this complexification of the NNs

The classical approach to the issue described in equation (3.5) consists in guessing a functional form for f :

$$f := f(x, \lambda), \quad (3.7)$$

where λ is a set of parameters which define the function f . The fitting is then performed by absolute minimization of some metric which measures the distance between the proposed function $f(x, \lambda)$ and the measured data D_i .

It is clear that such a method heavily depends on how well the functional for f has been chosen, thus introducing the risk of *biasing* the final result.

NNs can also be employed to infer the value of f : define the so-called *loss function* \mathcal{L} as:

$$\mathcal{L}(\lambda) = \sum_{i=1}^{N_{data}} \|D_i - NN(x_i, \lambda)\|^2. \quad (3.8)$$

where $NN(x_i, \lambda)$ denotes a NN *labeled* by x_i and whose output is defined by the set of weights λ .

The minimization of (3.8) becomes then a problem of great importance which needs its own discussion. We have to keep into account the fact the the values D_i are not the true values of the observable, but the shifted ones. Thus we do not want to actually reach the *absolute minimum* of the loss functions as for the classical case: given the fact that the NN can approximate essentially any function, reaching the absolute minimum of the loss would consist in reproducing the noise and not the underlying true value. Reproducing the noise when training a Neural Network is a phenomenon called *overfitting*.

In order to avoid such a problem a few techniques can be employed: in particular in the context of the NNPDF collaboration a *cross-validation minimization* method is employed, which works as follows. The training data are split in two sub-sets, one called training and the other validation set: the training set is the one over which we actually perform the minimization of the loss function; the other, the validation set, is used as a checking tool which tells us when to stop the minimization process. Figure (3.3) clearly represents the functioning of such a method.

When designing a NN for a specific fitting task, there is a number of *hyperparameters* to be chosen, some of which are listed below:

- layer number,
- activation function,

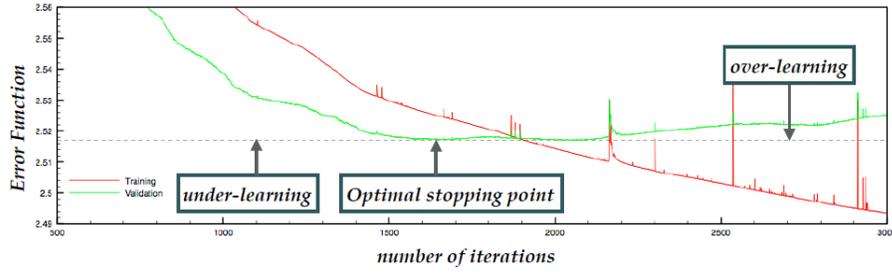


Figure 3.3: Graphical depiction of cross validation stopping point

- minimization strategy.

Hyperparameters are all those characteristics of the Neural Network which define its action before the actual fitting procedure. The choice of these hyperparameters can be carried out during a phase called *hyperparameter optimization*, whose process will be described for the case of the NNPDF Neural Network.

3.2 Neural Networks for PDF fitting: NNPDF

As mentioned in Chapter 1, hadronic cross sections are explainable in terms of the long range and the short range part of QCD. The short distance part, the PDFs, cannot be computed perturbatively and have to be fitted from experimental data.. First of all the specifics of the Neural Network of NNPDF will be described.

PDF fitting can be generally viewed as a really complex fitting procedure. The actual values of the measurements depend in a convoluted way on the PDFs; thanks to the factorization theorem we can view *any* observable as the result of the application of some functional to the PDFs. Take for example equation (1.26):

$$\sigma^{lh} = \sum_i \sum_f \int dx_i \int d\Phi_f q_i(x_i, Q_F^2) \frac{d\sigma^{lp \rightarrow f}(x_i, \Phi_f, Q_F^2)}{dx_i dQ_F^2}. \quad (3.9)$$

This could be simply made shorter by writing it down as

$$\sigma^{lh} = F(q_i(x_i, Q_F^2)), \quad (3.10)$$

where F represents the suitable *forward map* from the space of PDFs to the space of observables. Then, from a practical point of view, the implementation of such

a forward map becomes of great importance, topic which will be covered at the beginning of this section.

Furthermore the PDFs themselves need to obey to a lot of constraints coming from theoretical conclusions such as the ones briefly introduced in (1). As an example of these constraints one can consider the DGLAP evolution; in the following section we are going to introduce more constraints and elucidate how these are implemented in the fitting framework.

FK tables

FK tables are the numerical implementation of the observable map briefly mentioned in the above introduction. PDFs are parametrized at a reference scale Q_0 chosen here at $Q_0 = 1.65$ GeV. PDFs can be discretized on a grid of x_{grid} points:

$$\{f_i(x_\alpha, Q_0^2), i = 1 \dots N_f\}, \quad (3.11)$$

where the index i ranges over the flavours and the gluon, and α ranges over the grid of points on x -axis. Observables can be then written in terms of (3.11). A generic DIS observable takes the form:

$$F_I(x_J, Q_J) = \sum_{i=1}^{N_f} \sum_{\alpha=1}^{N_{grid}} \sigma_{i,\alpha}^{I,J} f_i(x_\alpha, Q_0^2), \quad (3.12)$$

where the tensor $\sigma_{i,\alpha}^{I,J}$ represents a pre-computed table for the observable F_I at a scale indexed by J . Hadronic observables on the other hand can be similarly written down as:

$$F_I(x_J, Q_J) = \sum_{i,j=1}^{N_f} \sum_{\alpha,\beta=1}^{N_{grid}} W_{i,j,\alpha,\beta}^{I,J} f_i(x_\alpha, Q_0^2) f_j(x_\beta, Q_0^2), \quad (3.13)$$

where clearly the pre-computed table W has to take into account the convolution of 2 different PDFs. The pre-computed tables σ for DIS and W for hadronic observables are the actual FK tables. It is important to notice that these take into account all theoretical information on an observable, especially enforcing DGLAP evolution onto the PDFs.

Hyeroptimization

Before deploying the actual fitting procedure the specifics of the NN used have to be defined. As said before these include qualitative characteristics that define the NN,

such as neuron number, number of layers, structure of the NN etc. The characteristics of the latest NNPDF 4.0 release can be summarized in the following table:

Parameter	NNPDF 4.0
Architecture	2-25-20-8
Activation function	Hyperbolic tangent
Optimizer	Nadam
Loss function	χ^2
Learning rate	2.6×10^{-3}
Free parameters	763
Max epochs	17×10^3

These hyperparameters are chosen during the so called *hyperparameter optimization* phase, whose exact functioning is described in [10].

NN architecture and flowchart

Having outlined the hyperparameters of the NN, we shall give an overview on the technical aspect of the NN. The Neural Network adopted by NNPDF is a fully connected neural network with the following structure:

As it can be seen from figure (3.4), the NN inputs are the points on the x -grid. The Neural Network is then the *direct* parametrization of the PDFs, which can be computed in evolution or in flavour basis.

$$\text{NN} : (x, \ln(x)) \mapsto \{f_i(x, Q_0^2)\}. \quad (3.14)$$

The training of the Neural Network can be summarised with the flowchart in figure (3.5). This flowchart briefly resumes the basic NNPDF procedure. The central block in particular refers to the actual fitting procedure: for each cycle the loss χ^2 is evaluated for a certain set of parameters. In order to make the discussion clearer we give here the definition of the loss function. The loss function in the context of NNPDF is defined as the chi square of the differences between NN predictions and observable values:

$$\frac{1}{N_{\text{data}}} \sum_{i,j=1}^{N_{\text{data}}} (D - T)_i C_{i,j}^{-1} (D - T)_j, \quad (3.15)$$

where D are the data values, T are the predictions of the NN and C is the *experimental covariance matrix*. We will dwell into the actual functioning of the minimization in the following section.

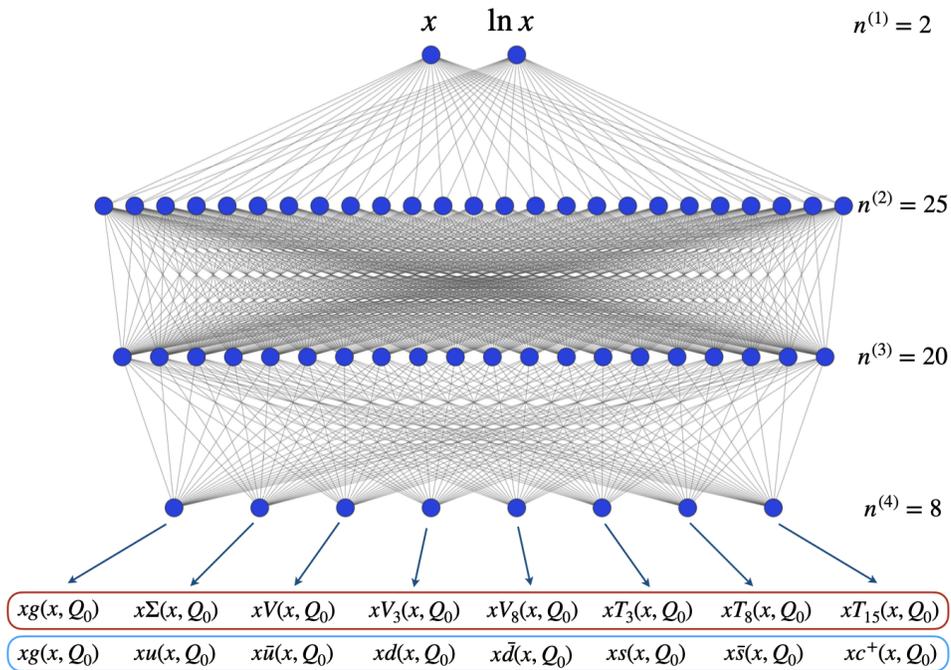


Figure 3.4: Neural Network adopted by NNPDF. Image taken from [7]

During each *optimization* iteration the set of weights of the NN is updated until a good stopping point is reached. The final three blocks of the flowchart (3.5) refer to the postfit selection; in particular the *APFEL evolution* refers to the DGLAP evolution which is deployed via the so called LHAPDF grids [8].

Monte Carlo replica approach

As already mentioned in the chapter related to error propagation, the NNPDF method for error propagation consists in a Monte-Carlo replica approach. From a technical point of view the method works as follows: as it can be seen in equation (2.16) we generate data replicas according to the experimental noise distribution. The actual generation of replicas is performed in the following way:

$$\mu = (1 + r^{norm} \sigma^{norm}) \left(y_0 + \sum_{p=1}^{N_{obs}} r^{p,sys} \sigma^{p,sys} + r^{stat} \sigma^{stat} \right), \quad (3.16)$$

where the various σ s refer to the various sizes of the errors, and r denote the actual R.V.s which are sampled when generating the shifts. In the end a fitting procedure

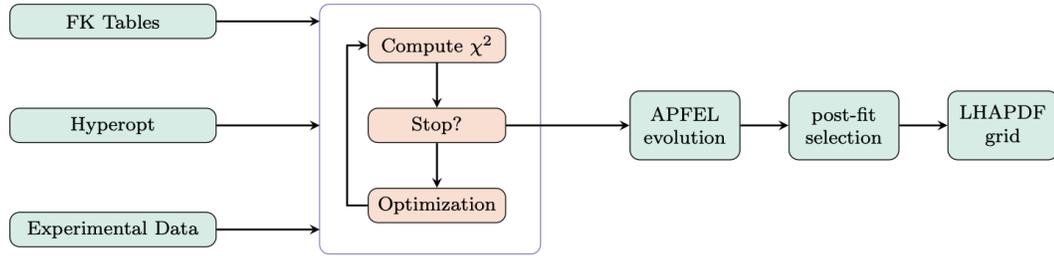


Figure 3.5: NNPDF flowchart

yields an ensemble of PDFs replicas which in turn yield a distribution for any kind of observable quantity.

Each PDF in particular is then obtained by the minimization of the χ^2 defined in equation (3.15). The way this minimization takes place is actually through the cross validation method, briefly introduced in the first section of the chapter and graphically shown in figure (3.3).

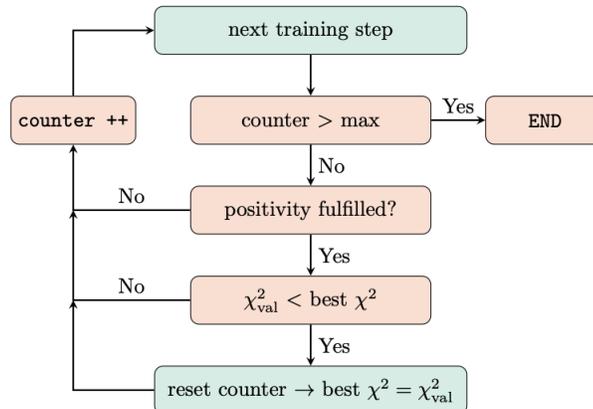


Figure 3.6: Stopping criterion employed by NNPDF. Image taken from [7]

In the NNPDF case the training/validation split amounts to keeping 75% of the data in the training set and the remaining 25% in the validation set. In figure (3.6) we are showing the functioning for the stopping criterion employed by NNPDF.

Parametrization basis

As said in Chapter 1 the DGLAP evolution equations are a set of coupled differential equations. These can be partially decoupled choosing a suitable basis of functions called *evolution basis*. In order to solve the DGLAP evolution equation one needs to parametrize the PDFs at the input evolution scale which is chosen at $Q_0 = 1.65$ GeV. Specifically the parametrized combinations are the following ones:

$$\begin{aligned}
\Sigma &= u + \bar{u} + d + \bar{d} + s + \bar{s} + 2c, \\
T_3 &= (u + \bar{u}) - (d + \bar{d}), \\
T_8 &= (u + \bar{u} + d + \bar{d}) - 2(s + \bar{s}), \\
V &= (u - \bar{u}) + (d - \bar{d}) + (s - \bar{s}), \\
V_3 &= (u - \bar{u}) - (d - \bar{d}), \\
V_8 &= (u - \bar{u} + d - \bar{d}) - 2(s - \bar{s}), \\
T_{15} &= (u + \bar{u} + u + \bar{u} + u + \bar{u}) - 3(c + \bar{c}).
\end{aligned} \tag{3.17}$$

It can be seen that here not all 13 independent PDFs are fitted; this is due to the fact that it is reasonable to think that some PDFs are equal to 0 at the evolution scale 1.65 GeV, thus making the set of functions smaller. PDFs can obviously also be given in flavour basis: both possibilities have their own advantages and disadvantages, but the choice that has been taken is to use the evolution basis as standard. Given the choice of the basis as the evolution one, a *preprocessing* of the PDFs can be implemented in order to speed up the training procedure (see later). PDFs are explicitly written in terms of the NN as:

$$xf_k(x, Q_0|\lambda) = A_k x^{1-\alpha_k} (1-x)^{\beta_k} NN_k(x|\lambda), \tag{3.18}$$

where α and β represent the *preprocessing exponents* and λ collectively represent the neural networks weights. The independence of the result of the preprocessing parameters has been proven in [11].

Theoretical constraints

PDFs are subject to a number of theoretical constraints as said in the introduction of the chapter. In the following subsection the implementation of these in the framework will be inspected.

Sum rules Irrespectively of the choice of the basis, PDFs must obey the so-called *sum rules*. These are a consequence of momentum conservation and in evolution basis read:

$$\begin{aligned} \int_0^1 dx x(g(x, Q) + \Sigma(x, Q)) &= 1, \\ \int_0^1 dx V(x, Q) &= \int_0^1 dx V_8(x, Q) = 3, \quad \int_0^1 dx V_3(x, Q) = 1. \end{aligned} \quad (3.19)$$

This set of 4 equations imposes constraints on the PDFs. These must be valid for each value of Q but given their validity at any Q_0 scale, DGLAP equations ensure their validity at any other Q dQ_0 . Recalling (3.18) these sum rules are imposed by fixing the normalization constants A_k which in this case namely are $A_g, A_V, A_{V_3}, A_{V_8}$.

Positivity of PDFs and observables Cross sections are observables whose meaning is that of a probability, thus they are all positive. PDFs beyond leading order on the other hand cannot be treated as probability density functions; in particular this character of PDFs is encoded by the specific renormalization scheme chosen. It has been proven in [9] that in the $\overline{\text{MS}}$ scheme PDFs have to be positive. The positivity of PDFs is enforced by introducing a penalty system: the loss function is augmented by a factor depending on positivity

$$\chi^2 \rightarrow \chi^2 + \sum_{k=1}^8 \Lambda_k \sum_{i=1}^{n_i} \text{Elu}_\alpha(-\tilde{f}_k(x_i, Q^2)), \quad (3.20)$$

where \tilde{f} refers to the PDFs in flavour basis, $Q^2 = 5\text{GeV}^2$ and Λ_k are Lagrange multipliers. The functions Elu is defined as follows:

$$\text{Elu}_\alpha(t) = \begin{cases} t & \text{if } t > 0 \\ \alpha(e^t - 1) & \text{if } t < 0. \end{cases} \quad (3.21)$$

The x_i points in (3.20) are chosen logarithmically spaced between $5 \cdot 10^{-7}$ and 10^{-1} and then linearly spaced between 0.1 and 0.9. The parameter in (3.21) is $\alpha = 10^{-7}$: it can be seen from the equation that the penalty is proportional to both the absolute value of the PDF and to the Lagrange multiplier.

Integrability conditions and postfit PDFs have to be integrable functions, condition which is related to the small- x behaviour. From equation (3.19), it can be seen

that sum rules imply the following:

$$\begin{aligned} \lim_{x \rightarrow 0} x^2 g(x, Q^2) &= \lim_{x \rightarrow 0} x^2 \Sigma(x, Q^2) = 0 \quad \forall Q, \\ \lim_{x \rightarrow 0} V(x, Q^2) &= \lim_{x \rightarrow 0} V_3(x, Q^2) = \lim_{x \rightarrow 0} V_8(x, Q^2) = 0 \quad \forall Q. \end{aligned} \quad (3.22)$$

From other results coming from standard Regge theory it can be also seen that non-singlet combinations T_3 and T_8 have to abide by similar conditions. The penalty system for integrability takes this structure:

$$\chi^2 \rightarrow \chi^2 + \sum_{k=1}^8 \Lambda_k \sum_{i=1}^{n_i} \left[x f_k(x_{\text{int}}^i, Q^2) \right]^2, \quad (3.23)$$

where the further label on the x points comes from the fact that we are only interested in the small x region. Finally also after training the integrability is imposed through a post-fit selection criterion which consists in discarding the PDFs which do not satisfy the following:

$$\sum_{i=1}^{n_i} \left| x_{\text{int}}^i f_k(x_{\text{int}}^i) \right| < \frac{1}{2}. \quad (3.24)$$

Chapter 4

Methodology Validation: Closure Test

This chapter is devoted to the accurate description of the framework of this thesis, which is the methodology validation of PDF determination. By methodology validation we mean a way which checks if the chosen method properly works in dealing with the assigned problem. In particular in this case we will be talking about the *closure test*.

4.1 Closure test mechanics

In this section we want to explain in detail what a closure test is. Since its first introduction in the context of NNPDF in [6], the closure test has been used to analyze the efficiency and accuracy of the NN in propagating uncertainties. The main idea behind a closure test is to test the NN behaviour in a controlled setting, in the sense that the true underlying value of each observable is known.

In a realistic situation the experimental collaborations provide us with the *central value* y_0 for an observable and an associated *covariance matrix* C_{exp} . As already mentioned, from a statistical point of view this set of information has the following meaning:

$$y_0 = f + \eta. \tag{4.1}$$

The true underlying value is called f and the experimental error is encoded in the random shift η , whose probability distribution is a multi-variate normal with mean 0

and covariance matrix given by the C_{exp} , that is $\mathcal{N}(0, C_{\text{exp}})$.

In a normal situation the value of f is not known. The starting point of a closure test is to *choose* a value for f and set it as underlying truth. From a practical point of view this means choosing a true value for the PDFs themselves, which are used to compute *any* observable. Knowing the true value f makes it possible to generate pseudo-data which resemble the realistic experimental measurements, sampling from equation (4.1). In the NNPDF jargon the various steps of pseudo-data generation are denoted as follows:

- Level 0 data: a set of underlying true PDFs is chosen. These are then used to compute the true value f of any observable, called *level 0 data*.
- Level 1 data: the level 1 data are the ‘copy’ of the experimental central values. Equation (4.1) is used in order to generate these and the random noise is sampled as said above.
- Level 2 data: as in any standard NNPDF fitting procedure the pseudo-data replicas are generated starting from the central values (see chapter 3).

Given the fact that we can generate multiple instances of level 1 data starting from the same set of level 0 data, we can perform several fitting procedures which will yield several samples of PDFs, to be compared to the underlying truth.

4.2 Statistical test

Having roughly defined the idea and framework behind a closure test the main problem is building a good statistical analysis of the results. We will start from the previously adopted figure of merit, following the steps in [13].

Let us start by defining an *output error* of the closure test in the following way:

$$E_{\text{out}} = \frac{1}{N_{\text{data}}} (O'(u_*) - y'_0)^T C_{\text{exp}} (O'(u_*) - y'_0). \quad (4.2)$$

This figure of merit is defined in accordance with the standard figure of merit for the normal tests, the χ^2 -loss previously defined. The prime in the equation above indicates the fact that for testing the goodness of fit we are using data which were not included in the training set. Even if we could use the same set also for testing we want to check to generalization power of the NN itself. In order to keep all

information possible the ideal situation would be to have testing and training set statistically independent:

$$\text{Cov}(y_0, y'_0) = \begin{pmatrix} C_Y & 0 \\ 0 & C_{Y'} \end{pmatrix}. \quad (4.3)$$

Just to be clearer the fake central values y'_0 are defined as usual:

$$y'_0 = f' + \eta', \quad (4.4)$$

where η' has to be regarded as a R.V. to be sampled from the same distribution of the level 1 shifts.

In order to have a better idea of the practical implementation of such an analysis, it is better to specialize the notation, which is going to be adopted through the rest of the work.

- η^l : this refers to the instance l of the noise generating level 1 data where l ranges from 1 to N_{fits} . N_{fits} samples of noise will yield N_{fits} samples of level 1 data defined as:

$$y_0^l = f + \eta^l. \quad (4.5)$$

- ϵ_r^l : this refers to instance r related to fit l of level 2 noise. r ranges from 1 to N_{rep} . Level 2 data samples are then indexed in the following way:

$$\mu_r^l = y_0^l + \epsilon_r^l. \quad (4.6)$$

- Given the two definitions above it is natural to define the best model fitted from each replica in the following way:

$$u_{*,r}^l := \text{best model obtained from fitting replica } r \text{ of fit } l. \quad (4.7)$$

- Testing central values y_0'' : the out of sample central values which appear in the definition of E_{out} are defined similarly to the central values for the fits:

$$y_0'' = f' + \eta'' \sim \rho. \quad (4.8)$$

in accordance with the standard fitting procedure. In a real case scenario even the out of sample set would be a collection of experimental central values.

The output error defined in (4.2) has to be regarded as a random variable, thus a sample of such a R.V. will inherit all the indices above said. This means that for each fit and for each replica we can define an error:

$$E_{out}(\eta^l, \epsilon_r^l, \eta^l) := \frac{1}{N_{\text{data}}} (O'(u_{*,r}^l - y_0^l))^T C'_{\text{exp}} (O'(u_{*,r}^l - y_0^l)). \quad (4.9)$$

For each fit l we can then use as figure of reference the mean across replicas:

$$E_{\text{out}}^l = \sum_r \frac{1}{N_{\text{rep}}} E_{out}(\eta^l, \epsilon_r^l, \eta^l). \quad (4.10)$$

It is implicit that N_{rep} does not carry any dependence on the fit index l , which practically means that each fit should have the same number of replicas.

Let us further split this equation into quantities of interest. The starting point is the following expression, which is just a rewriting of (4.10):

$$E_{\text{out}}^l = \frac{1}{N_{\text{data}}} \times \sum_r \frac{1}{N_{\text{rep}}} \left[(O'(u_{*,r}^l) - f')^T C'_{\text{exp}} (O'(u_{*,r}^l) - f') + (f' - y_0^l)^T C'_{\text{exp}} (f' - y_0^l) + 2(O'(u_{*,r}^l) - f')^T C'_{\text{exp}} (f' - y_0^l) \right]. \quad (4.11)$$

In this way we are highlighting the important components of the error, which are going to be analyzed in the following part. In order to further simplify this expression, following [13], we take the mean over the l index. From a practical point of view this means generating multiple level 1 data instances then deploying several fitting procedures. This is what is known as a *multiclosure test*.

$$E_{\text{out}} = \sum_l \frac{1}{N_{\text{fits}}} E_{\text{out}}^l \quad (4.12)$$

This averaging operation removes the cross term present in (4.11), leaving us with only the following:

$$E_{\text{out}} = \sum_l \frac{1}{N_{\text{fits}}} E_{\text{out}}^l = \sum_l \frac{1}{N_{\text{data}} N_{\text{fits}}} \times \sum_r \frac{1}{N_{\text{rep}}} \left[(O'(u_{*,r}^l) - f')^T C'_{\text{exp}} (O'(u_{*,r}^l) - f') + (f' - y_0^l)^T C'_{\text{exp}} (f' - y_0^l) \right] \quad (4.13)$$

A further thing to notice is the fact that the last term in (4.13) is simply noise

related, and its expected value is 1 given the definition of y'_0 . This yields

$$E_{\text{out}} = \sum_l \frac{1}{N_{\text{fits}}} E_{\text{out}}^l = \sum_l \frac{1}{N_{\text{data}} N_{\text{fits}}} \times \\ \times \sum_r \frac{1}{N_{\text{rep}}} \left[(O'(u_{*,r}^l) - f')^T C'_{\text{exp}} (O'(u_{*,r}^l) - f') + 1 \right] \quad (4.14)$$

in the limit of infinite fits. This leaves only the first term, which can be further decomposed:

$$(O'(u_{*,r}^l) - f')^T C'_{\text{exp}} (O'(u_{*,r}^l) - f') = \\ = (O'(u_{*,r}^l) - \sum_r [O'(u_{*,r}^l)])^T C'_{\text{exp}} (O'(u_{*,r}^l) - \sum_r [O'(u_{*,r}^l)]) + \\ + (f' - \sum_r [O'(u_{*,r}^l)])^T C'_{\text{exp}} (f' - \sum_r [O'(u_{*,r}^l)]), \quad (4.15)$$

where we have made use of the fact that if reinserted in the general expression, the cross term vanishes. The two expressions added together are respectively called *bias* and *variance*. Just to be explicit these read:

$$\text{Bias}^l := (f' - \sum_r [O'(u_{*,r}^l)]) C (f' - \sum_r [O'(u_{*,r}^l)]) =: \Delta_B^l C \Delta_B^l \quad (4.16)$$

and

$$\text{Variance}^l := \sum_r \left[(O'(u_{*,r}^l) - \sum_j [O'(u_{*,j}^l)])^T C'_{\text{exp}} (O'(u_{*,r}^l) - \sum_j [O'(u_{*,j}^l)]) \right] \quad (4.17)$$

$$=: \sum_r \Delta_{V,r}^l C \Delta_{V,r}^l. \quad (4.18)$$

We want to underline the implicit definitions in the above equation:

$$\Delta_{V,r}^l = (O'(u_{*,r}^l) - \sum_j [O'(u_{*,j}^l)]), \\ \Delta_B^l = (f' - \sum_r [O'(u_{*,r}^l)]), \quad (4.19)$$

which are going to be useful in the next paragraphs.

These quantities are essentially chi squared values calculated for different sets of points. The bias represents how far the central predictions are from the underlying truth, while the variance represents the spread of each fit replicas around their central

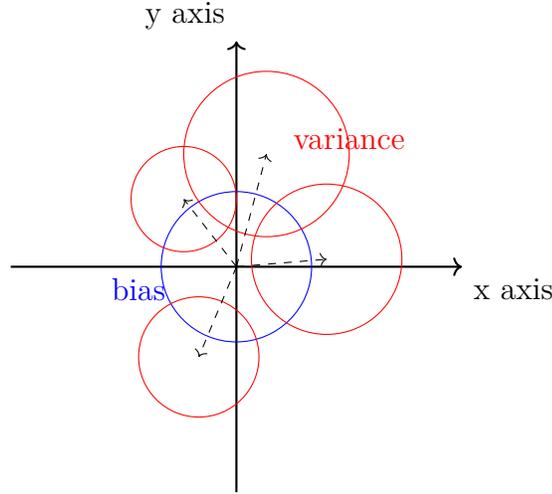


Figure 4.1: Graphical representation of bias-variance tradeoff. The blue circle represents the mean squared distance between central values and underlying truth. The red circles represent the mean square distance between central values and replicas.

value. The interplay between these two quantities can be used in order to understand whether a result is truly good or just ‘good-looking’ as it is graphically shown in figure (4.1):

The image shows the ideal situation: the blue circle represents the mean squared distance from the true underlying value, while the red circles represent the mean squared distances of each replica around the mean of each fit. The situation is ideal since the radii of the circles are of similar magnitude.

The figure of merit introduced to check consistency between the spreads of these quantities is the *square root bias variance ratio*, defined in (4.20).

$$\sqrt{R_{bv}} := \sqrt{\frac{\sum_l \text{Variance}^l}{\sum_l \text{Bias}^l}}. \quad (4.20)$$

If $\sqrt{R_{bv}} = 1$ then the NN is faithfully delivering uncertainties. For computational time reasons the number of samples of fits l is low (around 25 replicas) thus R_{bv} could be affected by great oscillations. Taking the square root of the quantity diminishes this effect, making the figure of merit more stable.

Chapter 5

Results

This final chapter is devoted to the results of the thesis. First of all we are going to show the problem which arose with the previous closure test methodology; after that we are going to introduce a new formalization chosen for the closure test which constitutes the first part of the results. After that we are going to dwell into the topic of inconsistent closure tests, the second part of this work.

5.1 Consistent closure test results

In this section we are going to show the results for the consistent closure test performed with the standard NNPDF method. First of all we are going to elucidate the technical details of the procedure.

Numerical setup

In each multiclosure test there is a number of defining free parameters that need to be chosen. These are mainly:

1. Number of replicas for each fit
2. Number of fits
3. Choice of underlying true value
4. Splitting between training data and out of sample testing data

Concerning point 1 and 2 the chosen numbers were ~ 25 fits, so 25 instances of level 1 data with 100 replicas each. The choice for these numbers comes from previous studies regarding the stability of the progressive mean of bias/variance ratio, which has been here repeated.

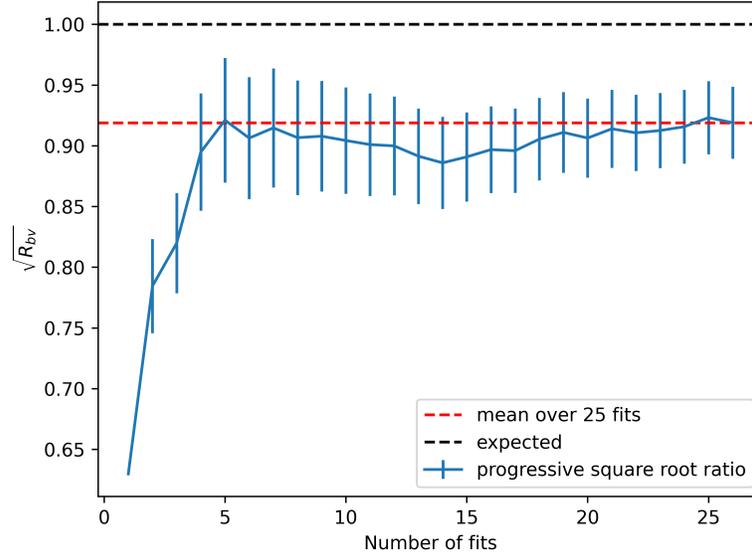


Figure 5.1: Progressive square root ratio. The errorbars are computed as the progressive standard deviation

Figure 5.1 shows that after 25 fits the square root ratio stabilizes on a value which in this case is 0.92. This plot shows that the chosen values for the closure tests guarantee stability of this figure of merit.

The underlying true value for the PDFs was chosen arbitrarily. This was also based on previous studies regarding the independence on the choice of underlying truth, which have not been repeated in the context of this thesis. Results can be found in [20]. In particular we choose the central set of PDFs of a previous fitting procedure in order to test the NN in a realistic context, which ensures to not have catastrophic outcomes related to the physical constraints which the PDFs must obey to.

The final point is the most delicate one, since it brings us to the core of the problem which affects the bias variance ratio. The splitting between in and out of sample of the data is useful since we want to test the performance of the methodology on data

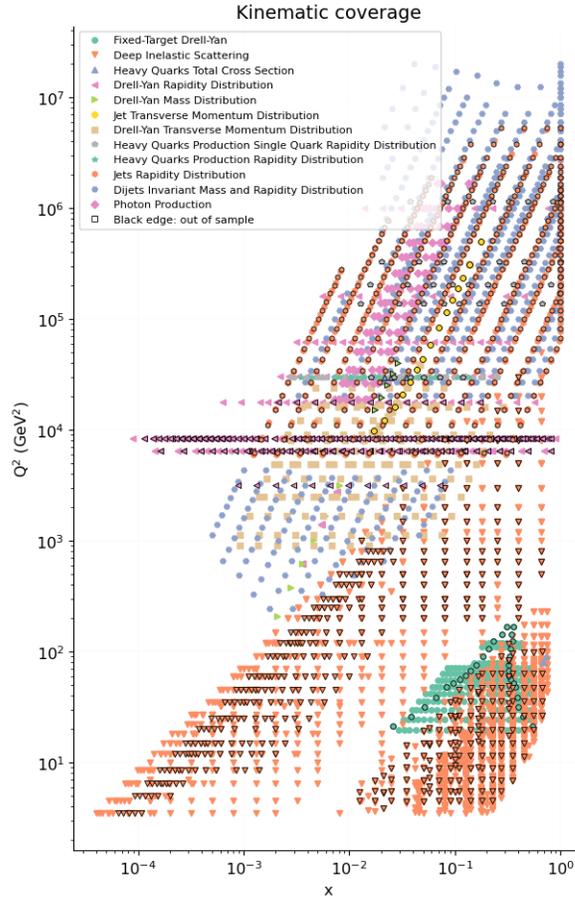


Figure 5.2: Kinematic coverage plot. Highlighted datasets are the out of sample ones comprised in the testing set

which was not included in the fit. This strategy is employed in order to simulate the performance of the NN on unseen data, so the power of the NN to generalize the information. In order to avoid biasing the result, the split between training and testing set must be made in the most homogeneous way possible.

The choice for such a split has been taken from the strategy of K-Folding, which consists in an automated choice for splitting. More details on the definition of the K-folding procedure can be found in [16].

In the following section we are going to present the results related to a global Consistent Closure Test. The global adjective refers to the fact that we are making use of the whole NNPDF 4.0 dataset, since we are also going to show results for smaller sets of data.

As it can be seen in Fig. (5.1), the square root ratio stabilizes on a value of 0.92. While this could be considered a good approximation to 1, the error bars computed as the progressive standard deviation of the sample of the ratio itself show that this final value should not be considered compatible with the expected one.

The inconsistency between the found result and the expected one needs to be properly addressed. Let us recall the notation for the consistent closure test: l denotes the index of the fit and r denotes the replica number. The raw output of a multiclosure test is a sample of PDFs u indexed in the following way:

$$u_r^l := \text{best PDF set of fit } l \text{ replica } r. \quad (5.1)$$

Since the analysis of the performance is performed in data space, these PDFs replicas yield the following predictions for observables:

$$\mu_{r,i}^l := O^i(u_r^l) \quad (5.2)$$

where O is the forward map and i indexes the specific observable. Let us for now consider a single observable and a single fit: this means fixing the indices $l = L$ and $i = I$.

The prediction $\mu_{r,I}^L$ can be considered a R.V. of which we have a sample, indexed by the replica number r . It is obvious though that we do not have any information regarding the distribution of this variable, apart from the fact that we have a sample for it.

The problem in the bias variance ratio as previously defined is that we are mixing information related to the distribution of experimental measurements with the distribution of the output of a NNPDF fit.

In particular this has the following effect on the bias and the variance: while they are defined essentially as χ^2 quantities they do not have any property related to the generalized chi square of a multivariate gaussian: this is because we are using as set of weights the experimental covariance matrix, which has no relation to the output distribution of the multi-closure test.

This can be easily seen by computing separately the bias and the variance for an ensemble of datasets. In figure (5.3) it can be clearly seen that there is no trend of the bias and the variance with respect to the dataset size: if the bias and the variance were actually χ^2 their expected values should both follow the line $y = x$

This problem requires a new formalization for the closure test figure of merit, explained in the following section.

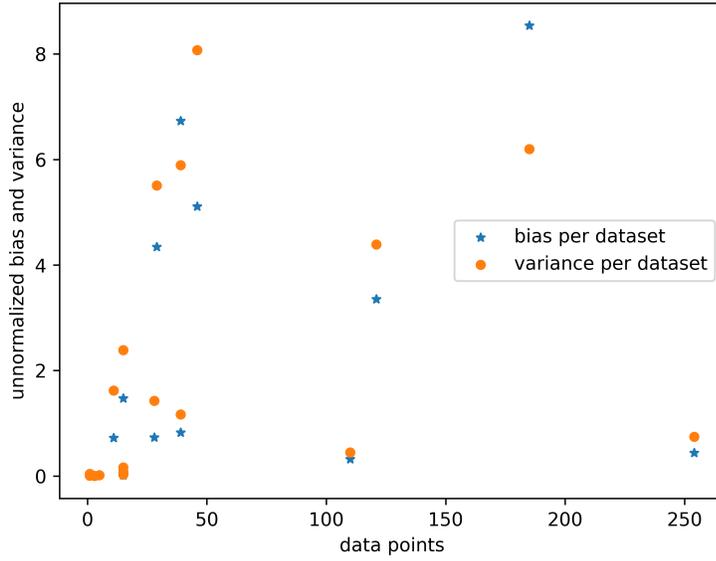


Figure 5.3: Bias and variance means for different datasets vs number of data points for each dataset

5.2 Closure test revisited

In this work the closure test formalism has been changed. The fitting operations and setup have been kept completely equal as already described, but the results have been analyzed in a different way. The main idea behind this proposal is avoiding the usage of the experimental covariance matrix, in order to avoid biasing the bias variance ratio.

The quantities of interest are going to be:

- $\Delta_{V,r,i}^l$
- $\Delta_{B,i}^l$

which are the bias and variance distances previously defined for observable i .

Checking the consistency in error propagation of the NN consists in checking whether the R.V. $\Delta_{B,i}^l$ is compatible with the R.V. $\Delta_{V,r,i}^l$.

First of all we tried to simply redefine the bias variance ratio figure of merit replacing the experimental covariance matrix with the *sample covariance matrix* of the samples themselves. From a practical point of view this means the following:

- Select an out of sample testing set
- Compute a sample covariance matrix C_{PDF} for each fit using as sample the instances of $\Delta_{V,r,i}^l$.
- Define the bias and the variance as:

$$B^l := \Delta_{B,i}^l C_{PDF,ij}^{-1} \Delta_{B,j}^l \quad (5.3)$$

$$V^l := \frac{1}{N_{\text{rep}}} \sum_r \Delta_{V,r,i}^l C_{PDF,ij}^{-1} \Delta_{V,r,j}^l. \quad (5.4)$$

- Compute the bias variance ratio as usual.

In this way the quantities Δ_B and Δ_V are correctly weighted. The problem though in this approach resides in the *inversion* of the PDF covariance matrix. Remember that we are considering the predictions of the PDFs replicas in data space: it is obvious that the correlation between them is induced by the forward map which we are using to compute theoretical predictions. Consider a simple but realistic case: suppose the out of sample testing set comprises two observables which are *almost* the same. This could mean that we have included in the testing set e.g. the structure function measured for two really similar values of x . It is then straightforward that the predictions for these two quantities will lie almost on a straight line.

Two linearly dependent R.V.s are 100% correlated, yielding a singular matrix, thus rendering its inversion impossible. In the context of this work we tried the following approach.

First of all in order to avoid problems related to dimensionality we considered the correlation matrix instead of the covariance one:

$$\text{Corr} = \begin{pmatrix} \frac{\text{Var}(\mu_1)}{\text{Var}(\mu_1)} & \frac{\text{Cov}(\mu_1, \mu_2)}{\sqrt{\text{Var}(\mu_1)\text{Var}(\mu_2)}} \\ \frac{\text{Cov}(\mu_2, \mu_1)}{\sqrt{\text{Var}(\mu_1)\text{Var}(\mu_2)}} & \frac{\text{Var}(\mu_2)}{\text{Var}(\mu_2)} \end{pmatrix}. \quad (5.5)$$

We then numerically found the diagonalizing change of basis matrix V and the set of eigenvalues λ . After this we delete the couples of eigenvalue-eigenvector if the eigenvalue λ is below a certain threshold. This yields a modified change of basis matrix V' and a new set of eigenvalues λ' . After this we can restore the original shrunked correlation matrix by the following operation:

$$\text{New Corr} = V'^T \text{diag}(\lambda') V \quad (5.6)$$

This new correlation matrix can be then inverted avoiding problems related to numerical instability of the inversion.

The problem with this approach is that the threshold on the eigenvalues does not guarantee stability: from figure (5.4) to figure (5.7) a series of plots of the $\sqrt{\frac{B}{V}}$ for different values of the threshold, which can be clearly seen to vary as the threshold increases.

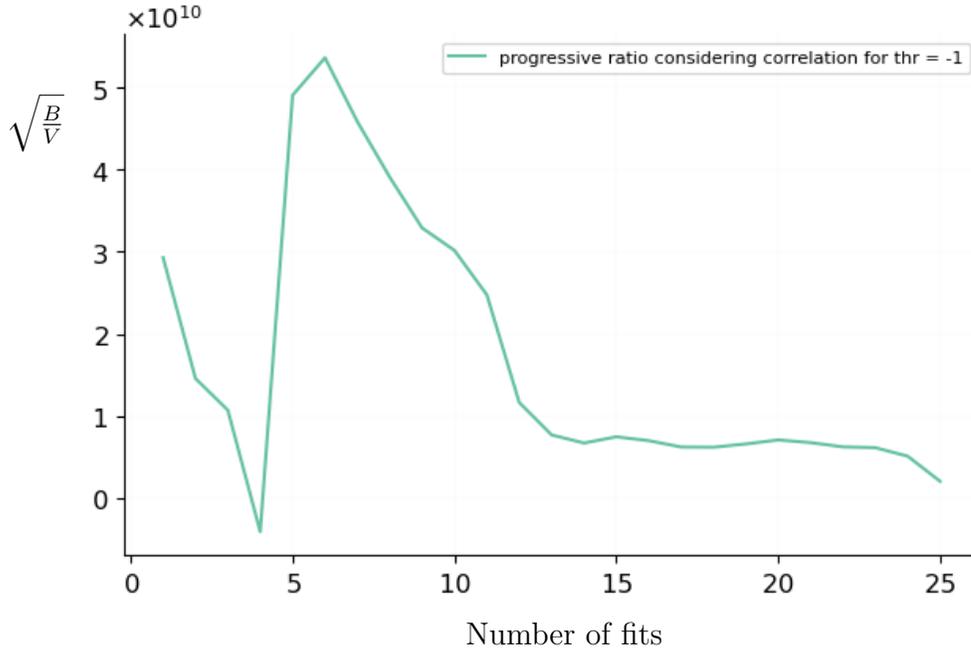


Figure 5.4: Bias variance square root ratio computed with regulated PDF covariance matrix. Threshold is -1

From the plots shown between figure (5.4) and figure (5.7) it is clear that we cannot approach the closure test with the proposed method keeping the correlations.

We can still check the liability of the NN in propagating uncertainties by looking at the *single* observables, without taking into account the correlations between them induced by the forward map. We are essentially going to make use of a kind of bias variance ratio once again as will be explained in the following section.

Define yet again the testing set as a collection of observables which have not been used in the training of the NN:

$$\text{Testing set} := \{\mu_i\} \text{ where } i = 1 \dots N_{\text{obs}}. \quad (5.7)$$

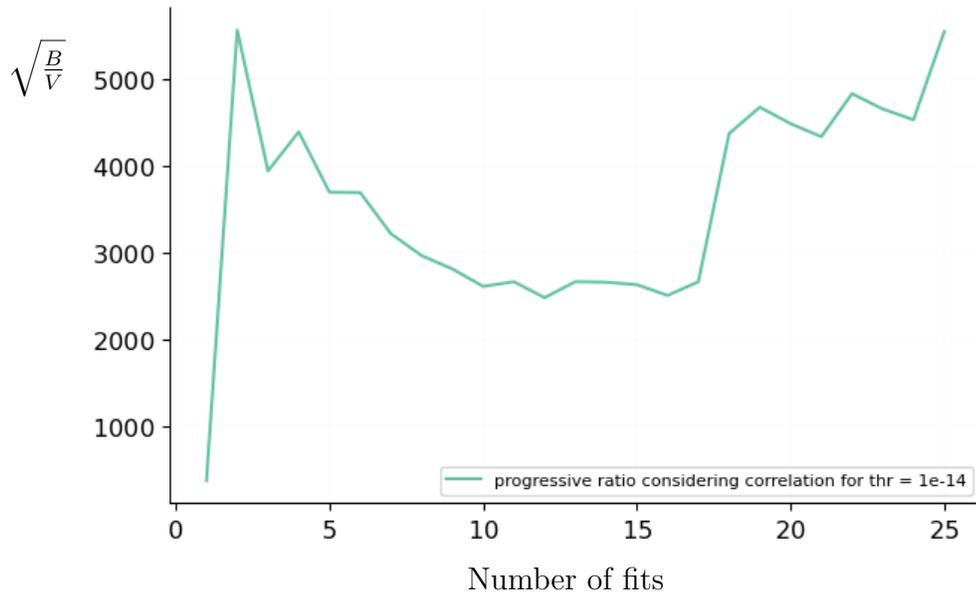


Figure 5.5: Bias variance square root ratio computed with regulated PDF covariance matrix. Threshold is 10^{-14}

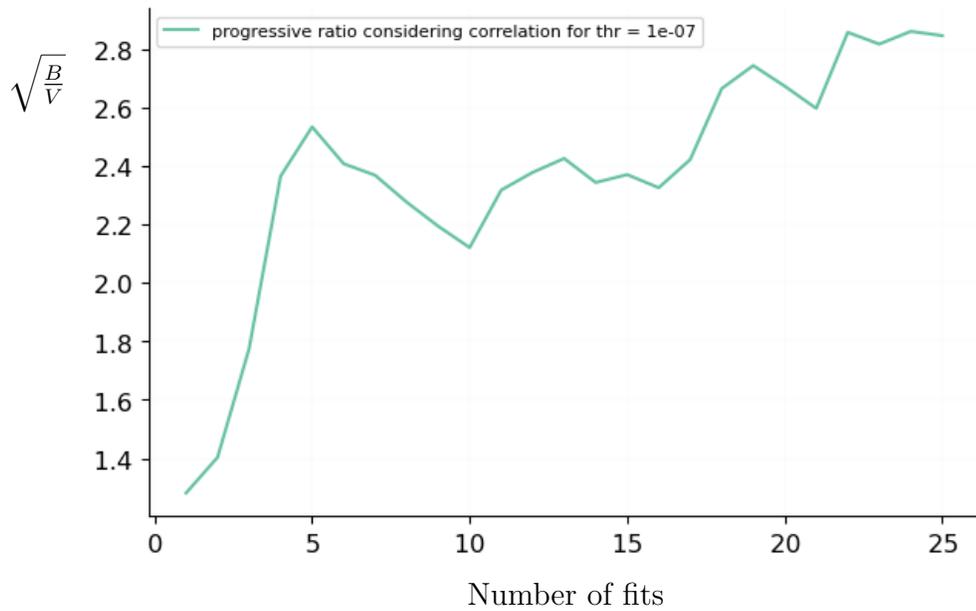


Figure 5.6: Bias variance square root ratio computed with regulated PDF covariance matrix. Threshold is 10^{-7}

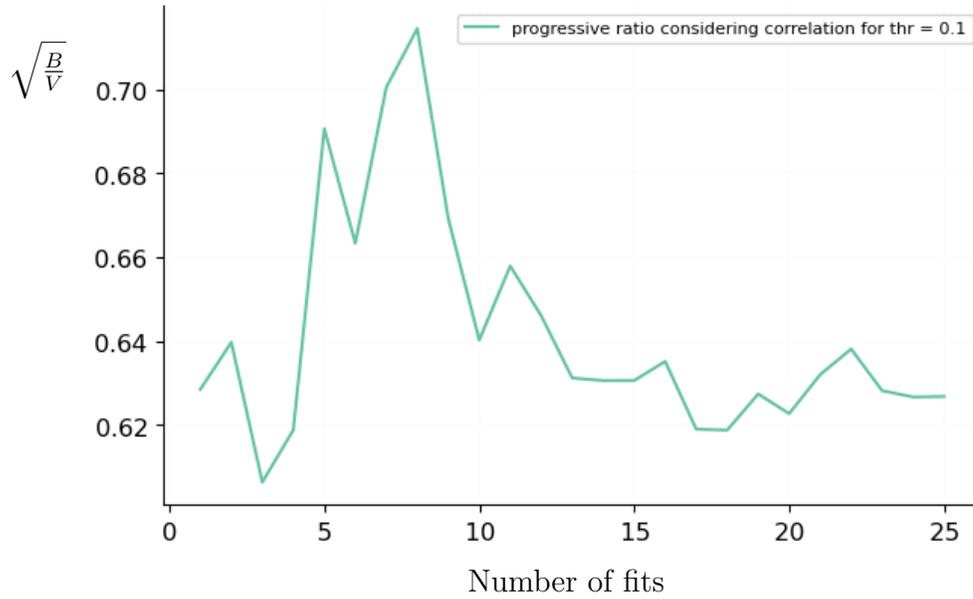


Figure 5.7: Bias variance square root ratio computed with regulated PDF covariance matrix. Threshold is 10^{-1}

We know that PDFs are gaussianly distributed thanks to a previous study conducted in [8]. Assuming linearity of the forward map we can still assume gaussianity in the distribution of the observables, thus we know that the R.V. Δ_V is gaussianly distributed:

$$\Delta_V^i \sim \mathcal{N}(0, \sigma_{pdf}^i), \quad (5.8)$$

where σ_{pdf} is the standard deviation of the sample.

Checking the NN consistency in propagating error consists in checking whether the quantity Δ_B^i is distributed according to:

$$\Delta_B^i \stackrel{!}{\sim} \mathcal{N}(0, \sigma_{pdf}^i), \quad (5.9)$$

which can be reformulated as:

$$\frac{\Delta_B^i}{\sigma_{pdf}^i} \stackrel{!}{\sim} \mathcal{N}(0, 1) \quad (5.10)$$

This change in the definition of the closure test gives us the possibility of checking point by point the validity of the algorithm, thus making it also possible to find the regions in (x, Q^2) space over which the NN performs better or worse. In the following section we are giving the new results for the Consistent Closure Test.

Results for consistent closure test

We can here show the results for the consistent closure test.

As said above the null hypothesis for testing states the equality between two distributions:

$$\frac{\Delta_B^i}{\sigma_{V^i}} \sim \mathcal{N}(0, 1), \quad (5.11)$$

which indicates the fact that the NN is correctly propagating uncertainty.

In order to show the results for the consistent test we are going to show first of all a histogram which plots the instances of the R.V. $\frac{\Delta_B^i}{\sigma_{V^i}}$ aggregating all the data together. This means not making any distinction between the various observables labelled by i .

The problem with this kind of histogram is that it does not keep into account the fact that all observables are inevitable correlated by the forward map. In an extreme case this could mean counting several times the same observable which would bias the distribution itself. We still show the histogram in force of the fact that the out-of-sample testing set has homogeneous characteristics given by the K-Folding procedure.

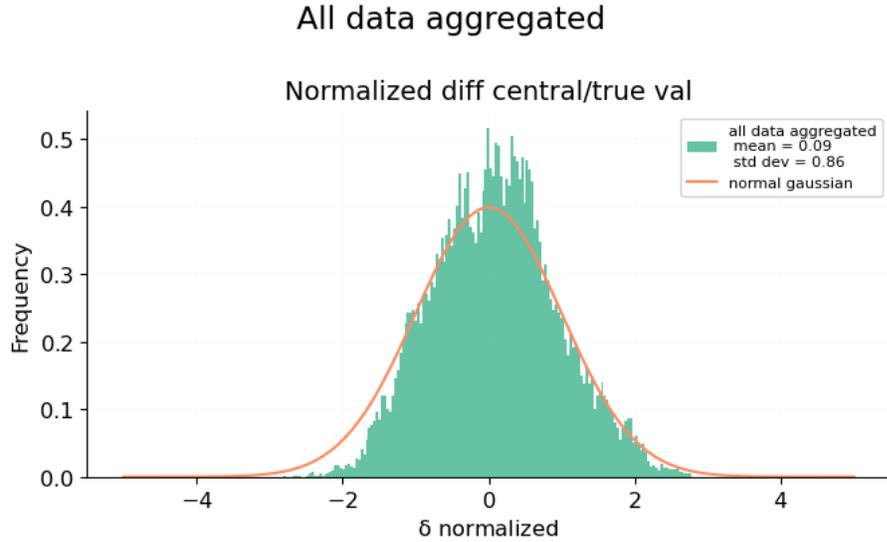


Figure 5.8: Results over all out of sample data for consistent closure test

The mean and the standard deviation of this histogram are:

$$\mu = 0.09, \quad (5.12)$$

$$\sigma = 0.86. \quad (5.13)$$

As it can be seen we are slightly over-estimating the uncertainty: in the ideal case the value of σ would be $= 1$, but, as said before, this could be related to the wrong weighting of some processes which slightly bias the histogram.

This figure of merit, while being rough, still gives an idea of the performance of the NN: in particular it will be interesting to look at this figure of merit in the context of the inconsistent closure tests.

In order to give an idea of the performance relating it also to the kind of process, we want to show the plot in figure (5.9): we plot each single data point in the (x, Q^2) plane also specifying the process type in the legend. The colorbar represents the *standard deviation* of the sample of the quantity $\frac{\Delta_B^i}{\sigma_{V^i}}$.

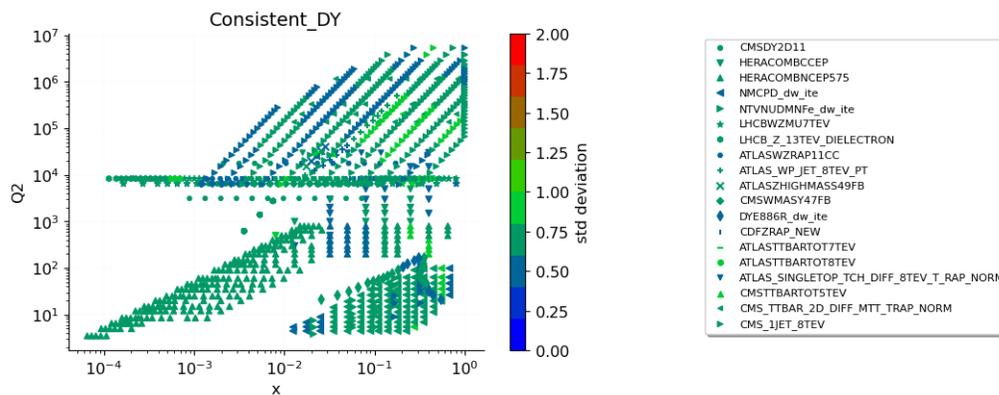


Figure 5.9: (x, Q^2) plane for out of sample testing set.

Also in this plot we can see that we are slightly overestimating the error given the presence of some point lying in the blue region, which corresponds to the interval $[0.4, 0.6]$, while no point lies in the symmetric brown region $[1.4, 1.6]$.

The proposed method of analysis is going to be adopted also for the inconsistent closure tests, which are the topic of the next subsection.

5.3 Inconsistent closure test

In the following subsection we are going to dwell into the topic of *inconsistent closure test*. The need for inconsistent closure tests arises when we want to simulate the situation in which the experimental collaborations have made a mistake in estimating sources of uncertainties affecting some observables. First of all then we are going to

define what is meant by inconsistent data, then we are going to present the results for a couple of cases of study which have been taken into consideration the context of this work.

Inconsistent data

Before talking about inconsistent closure tests we need to give the definition of inconsistent data. A measurement is said to be inconsistent if its nominal uncertainty is smaller than its real uncertainty. When we are talking about single measurements, thus scalar ones, this can only mean that the nominal standard deviation is smaller than the true one. On the other hand, when we are talking about correlated measurements, thus multi-dimensional ones, this can also mean a mistake in estimating the correlations between measurements has been made. In order to give a precise definition we are going to review the case of inconsistent data in the context of NNPDF fitting.

As said many times in the course of this work, also in the context of a normal fitting procedure the Monte Carlo replicas are generated by adding a stochastic noise to the central values. In order to relate this shift to the actual nominal uncertainties given by the experimental collaborations we need to properly define how the experimental covariance matrix is built. Taking a look at the definition in [14], the covariance matrix is built as:

$$\text{cov}_{ij} = \left(\sum_{k=1}^{N_{sys}} \sigma_{i,k} \sigma_{j,k} + F_i F_j \sigma_N^2 \right) + \delta_{ij} \sigma_{i,t}^2, \quad (5.14)$$

where F_i, F_j are the experimental central values, $\sigma_{i,k}$ are the N_{sys} correlated systematic uncertainties, σ_N is the total normalization uncertainty and $\sigma_{i,t}$ is the uncorrelated uncertainty.

We say that a set of data is not consistent if some of these errors have been wrongly estimated. In the context of an inconsistent closure test we can reproduce the situation in which some of these uncertainties have been underestimated: the final goal is to understand whether the NN procedure for fitting the PDFs *learns* this error and propagates it to the PDFs themselves. In order to give a more technical insight into the exact procedure of the inconsistent closure test, let us repeat the terminology for the closure test in a slimmer fashion adapting it to the inconsistent case.

A set of true PDFs is chosen, which is then used to compute the underlying truth, the level 0 data.

After this a set of level 1 data is generated sampling a noise from a known probability distribution in data space ρ . For each level 1 instance a set of level 2 instances is sampled adding a further noise to the level 1 data, this time sampling from a different distribution, ρ' .

$$\begin{aligned} \text{Lvl 1 data : } y_0 &= f + \eta \sim \rho, \\ \text{Lvl 2 data : } \mu &= y_0 + \epsilon \sim \rho'. \end{aligned} \tag{5.15}$$

The distribution ρ' comes exactly from modifying the covariance matrix defined in equation (5.14). Depending on the situation we chose some of the systematic uncertainties and rescaled them by a factor $\lambda \in [0, 1]$, being 1 the consistent case and 0 the extreme opposite in which we remove the systematic completely. This means then considering a modified covariance matrix defined as:

$$\text{cov}'_{ij} = \left(\sum_{k=1}^{N_{sys}} \lambda_k \sigma_{i,k} \lambda_k \sigma_{j,k} + F_i F_j \sigma_N^2 \right) + \delta_{ij} \sigma_{i,t}^2, \tag{5.16}$$

where the factor $\lambda_k = 1$ if we do not affect that uncertainty, and $\lambda_k < 1$ if we decide to affect that uncertainty.

Inconsistent closure test results

The goal of this study is understanding the impact of inconsistencies on the output of the Neural Network. Being the inverse problem really complex it is impossible to analytically understand in which regions of data space the resulting PDFs will yield problems, thus the only viable option is inserting inconsistencies in different datasets and processes in order to understand the NN response to the inconsistent input.

In order to understand this feature, the proposed method is the following: a set of data-points coming from a specific dataset is made inconsistent varying its error as explained in equation (5.15). An *inconsistent* closure test is deployed and then tested on a testing set set which has more or less the same characteristics as the training one. In particular it is important to include observables which closely match the inconsistent ones in the training, since we expect the NN to perform badly on these data-points. In the context of this work we have studied the impact of the inconsistency in three different cases:

- DIS-only fit.
- Drell-Yan inconsistencies.
- Jets data inconsistencies.

DIS inconsistent closure test

We first study the impact of inconsistencies on an only-DIS closure test. In this closure test we have used only DIS observables to perform both the training and the testing.

This can be seen as a preliminary study for the global tests since the Deep Inelastic Scattering observables are linear in the PDFs. Given the linearity at leading order of the forward map, we can expect the NN to propagate the errors learned in the inconsistent datasets to the out of sample testing set.

Choice of datasets and of uncertainties

In order to settle down the specifics for the inconsistent closure test, we need to choose which datasets we want to make inconsistent and which uncertainties we want to manipulate. We chose to make inconsistent the measurements coming from the HERA experiments [3], [4]. In particular we chose to insert inconsistencies in the following processes:

- inclusive DIS measurements of $e^\pm p$ collisions at $\sqrt{s} = 575$ GeV.
- inclusive DIS measurements of $e^\pm p$ collisions at $\sqrt{s} = 820$ GeV.
- inclusive DIS measurements of $e^\pm p$ collisions at $\sqrt{s} = 920$ GeV.
- inclusive DIS measurements of $e^\pm p$ collisions at $\sqrt{s} = 320$ GeV.

These datasets' fraction with respect to the whole number of data-points is $N_{\text{inc}} = 860$ against a total of $N_{\text{tr}} = 2576$ training data.

In second place we need to choose which uncertainties we want to modify. In order to measure the weight of the systematic uncertainties which we chose, we decided to use as a metric the trace of the covariance matrix. The trace of the covariance matrix

can be thought of as representing the ‘size’ of the overall uncertainty since:

$$\text{Tr}(\text{cov}) = \sum_{i=1}^D \lambda_i, \quad (5.17)$$

where λ_i are the eigenvalues of the covariance matrix. In the gaussian case the eigenvalues of the covariance matrix are the variances of the observables in the diagonal basis.

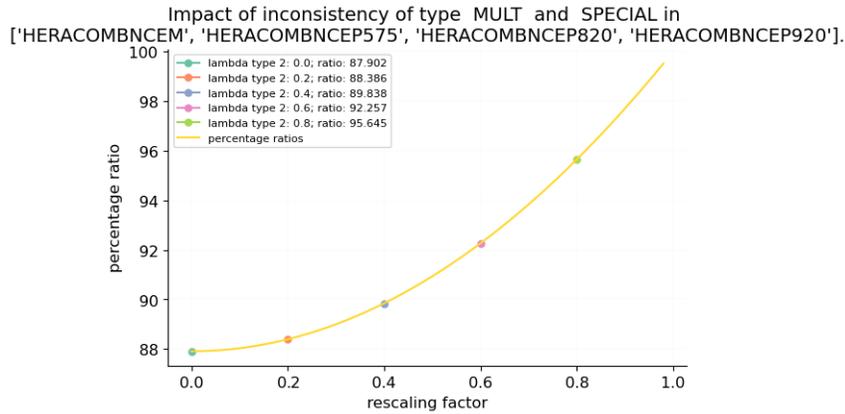


Figure 5.10: Trace variation vs λ

The plot in figure (5.10) shows the following ratio:

$$\frac{\text{Tr}(C(\lambda))}{\text{Tr}(C_{\text{exp}})}, \quad (5.18)$$

where the covariance matrix taken into consideration is the one related only to the inconsistent datasets, thus:

$$C = \begin{pmatrix} C_{\text{HERA}_1} & 0 & 0 & 0 \\ 0 & C_{\text{HERA}_2} & 0 & 0 \\ 0 & 0 & C_{\text{HERA}_3} & 0 \\ 0 & 0 & 0 & C_{\text{HERA}_4} \end{pmatrix}. \quad (5.19)$$

We could have shown the same plot using the ratio of the covariance matrix of the whole training dataset, but the impact on the trace variation is so small that it cannot be seen. Having defined all the premises, we can show the results of the DIS *inconsistent closure test*.

Results for inconsistent DIS fits

First of all we want to show the kinematic coverage of the training data, highlighting the inconsistent data points. As it can be seen the inconsistent data points cover a wide region in (x, Q^2) space, given also the fact that they constitute a third of the total training data. As already said in the initial part of this chapter related to the

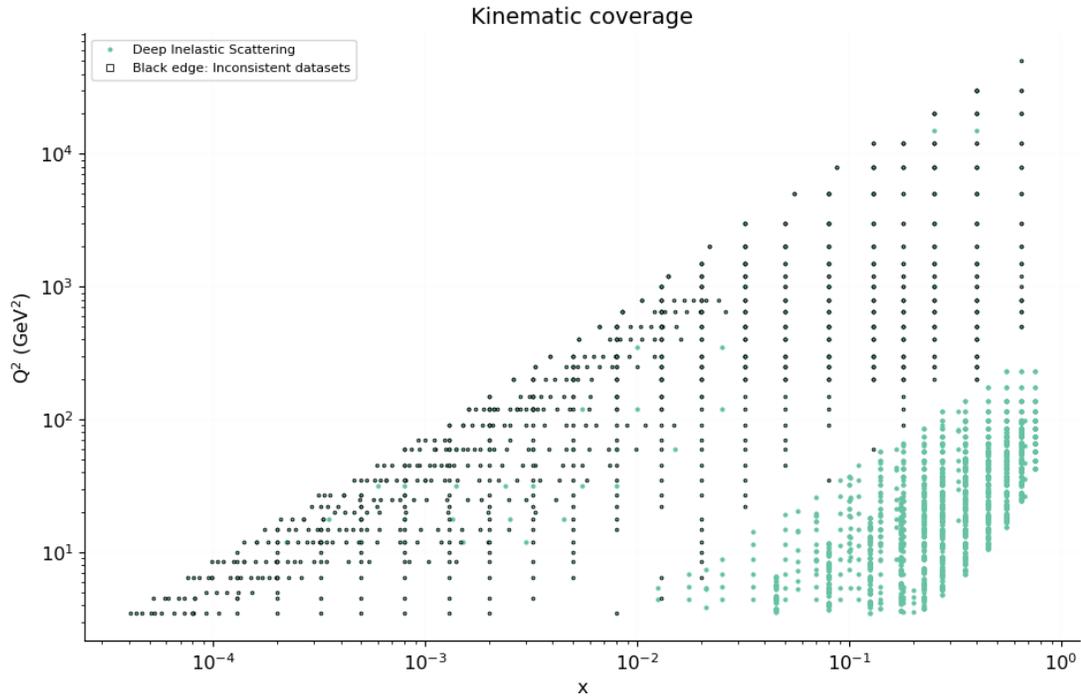


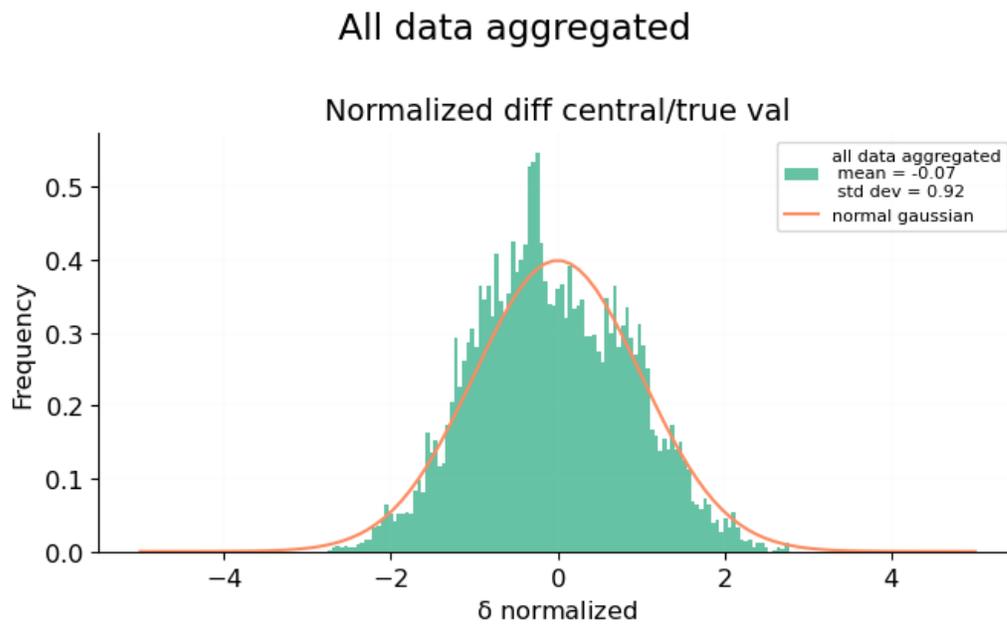
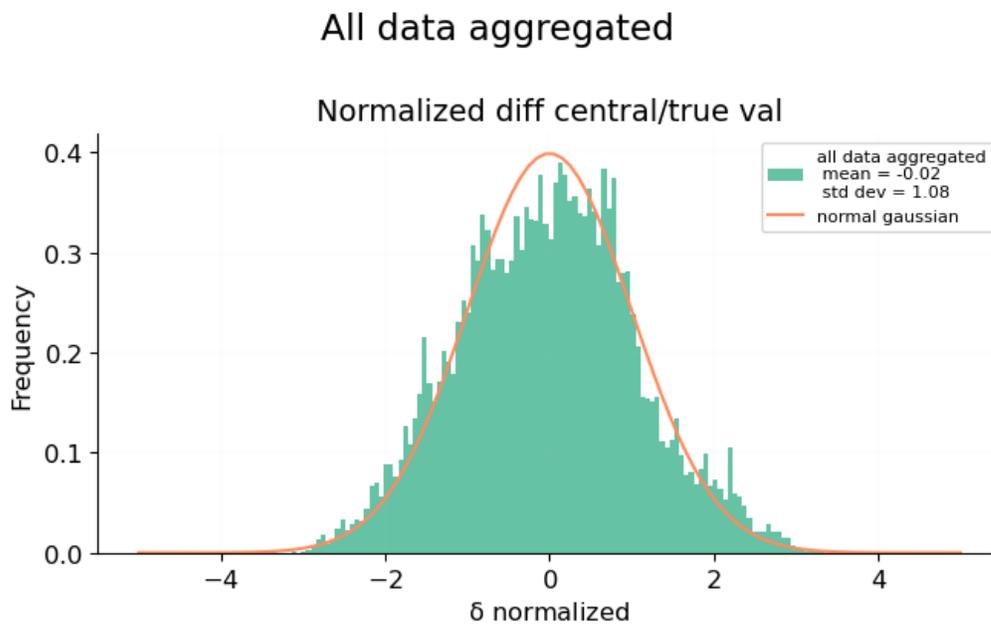
Figure 5.11: Inconsistency coverage

results, we are going to show the same plots and analysis proposed for the consistent closure test.

First of all we show the histograms of the $\frac{\Delta_B}{\sigma_V}$ for all the data in the out of sample test aggregated together.

It can be seen that as λ diminishes the shape of the histogram worsens, as it becomes wider. Qualitatively we can see that the inconsistency becomes really visible only when $\lambda = 0$, thus when we simulate the situation in which an experiment completely missed several sources of uncertainty.

More quantitatively we can also see how the mean and standard deviations of the histograms evolve with the inconsistency. The values are the following:

Figure 5.12: Histogram showing normalized Δ_B for consistent fitFigure 5.13: Histogram showing normalized Δ_B for inconsistent fit; $\lambda = 0.6$

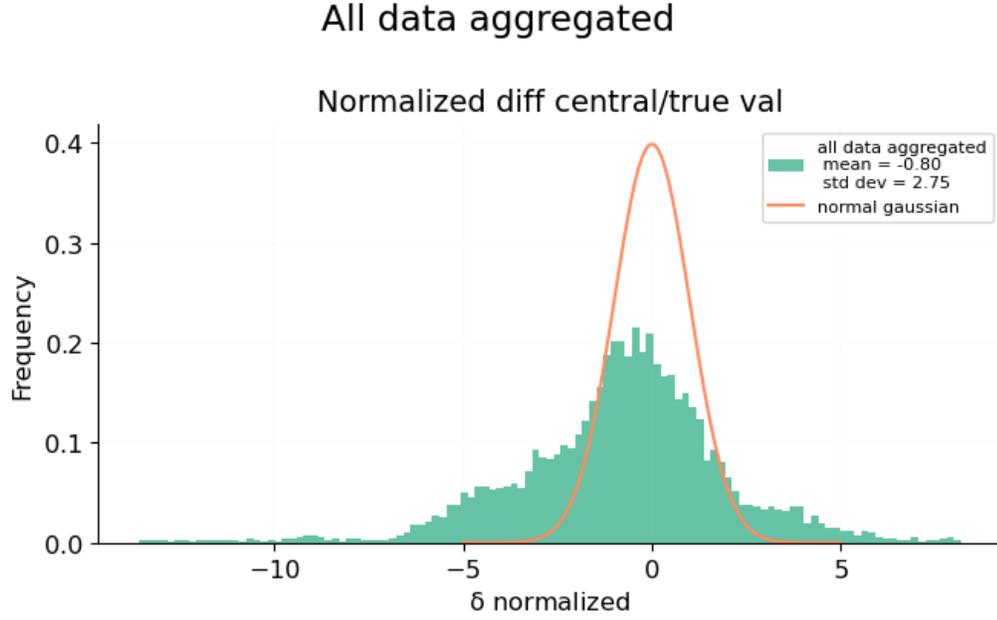


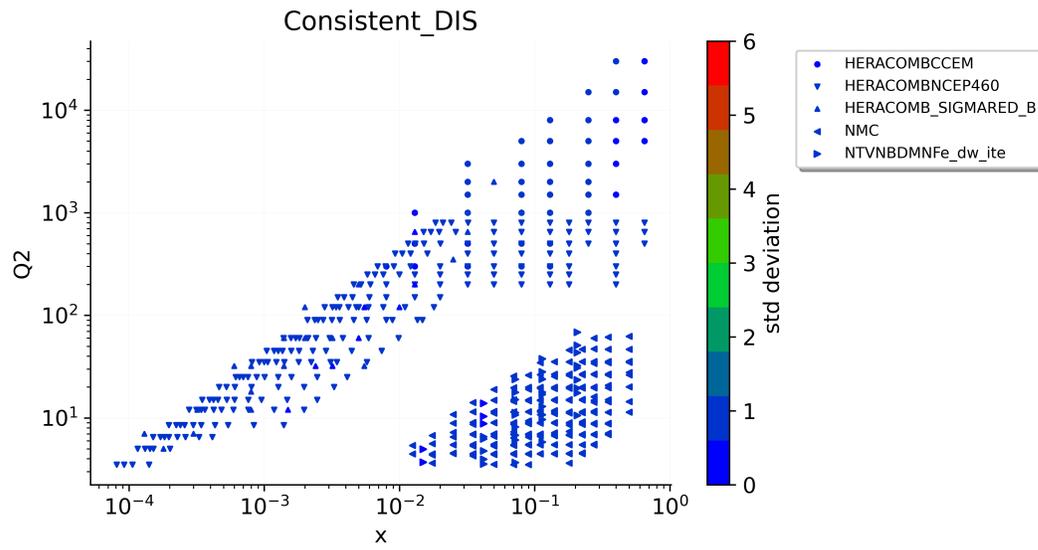
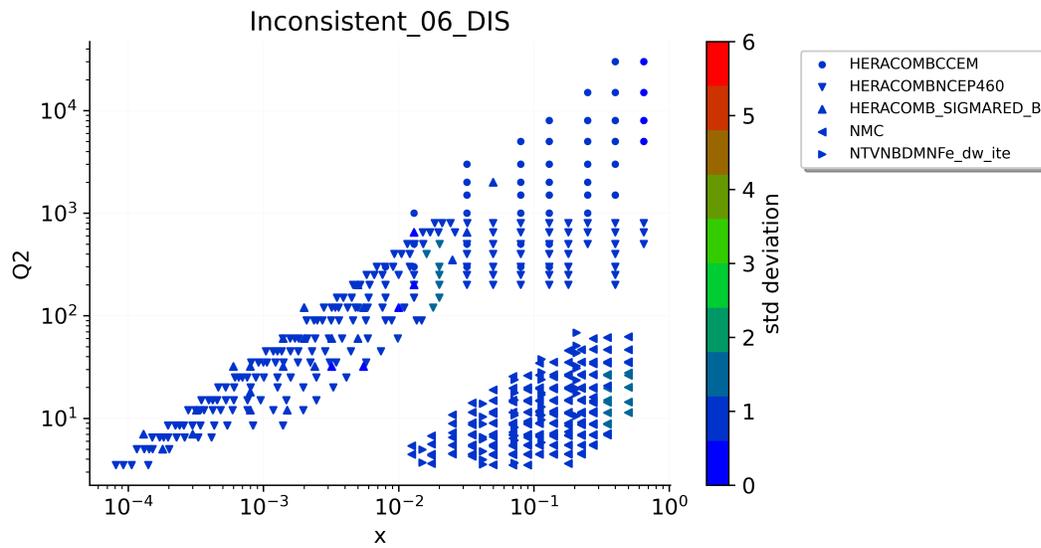
Figure 5.14: Histogram showing normalized Δ_B for inconsistent fit $\lambda = 0.0$

- Consistent test, $\lambda = 1$: $\mu = -0.07$ and $\sigma = 0.92$,
- Inconsistent test, $\lambda = 0.6$: $\mu = -0.02$ and $\sigma = 1.08$,
- Inconsistent test, $\lambda = 0.0$: $\mu = -0.80$ and $\sigma = 2.75$.

We can see that the values of σ for the intermediate inconsistent fit is as compatible with 1 as the value of the consistent fit itself. We could actually say that a smaller inconsistency would bring the NN to behave ‘perfectly’, thus yielding a value of $\sigma = 1$. Still we need to remark the fact that this figure of merit is surely biased by the correlations between data points. The final inconsistent fit for $\lambda = 0$ which is the extreme case points out that the NN is not able to absorb the inconsistency anymore and behaves corrupting the results.

On the other hand this histogram does not give any information on *which* particular observables are worsened by the inconsistency: in order to highlight this characteristic we can see the evolution of the heatmap of the consistent closure test until the inconsistent case. As previously said the colourbar represents the standard deviation of the sample of the R.V. $\frac{\Delta_B^i}{\sigma_{pdf}^i}$.

This series of heatmaps gives more detailed information related to the impact of

Figure 5.15: Standard deviation plot for consistent DIS. $\lambda = 1.0$ Figure 5.16: Standard deviation plot for inconsistent DIS. $\lambda = 0.0$

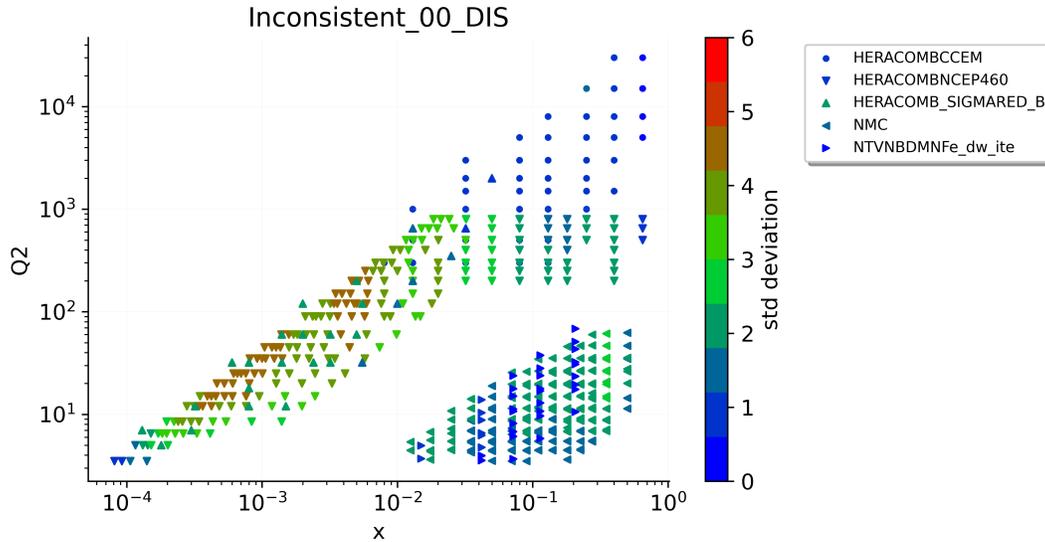


Figure 5.17: Standard deviation plot for inconsistent DIS. $\lambda = 0.0$

the inconsistency since we are not aggregating any set of data; in particular these heatmaps are telling us that the inconsistency placed in Neutral Current experiments in the training set affects the PDFs in such a way that mainly Neutral Current observables are predicted in the wrong way. More results are shown in the appendix which highlight this feature.

Conclusions for DIS inconsistent fits

The performed analysis suggests a couple of interesting patterns in the response of the methodology to the inconsistency in DIS data.

First of all we can see by looking at the trend of mean and standard deviation two effects: first of all the NN almost completely reabsorbs the inconsistency in the intermediate case, $\lambda = 0.6$. In second place in the extreme case, not only the standard deviation is worsened, but also the mean is shifted away from the expected value of 0. It is interesting to remark the fact that only with a great amount of inconsistency the effect is clearly visible.

Getting more specifically into which observables are the affected ones, it can be seen that these are the ones related to Neutral Current DIS experiments, which are the inconsistent ones in the training set. It can be also seen though that not all the Neutral Current observables are affected by the inconsistency. What is especially

interesting to notice is the fact that the observables in the same kinematic region as the one in which the inconsistency was inserted have been affected more than the rest.

These results related to the DIS inconsistent closure test show that the NN propagates the inconsistency from training data to testing data, affecting only the same kinds of processes which were made inconsistent in the training data.

Drell Yan inconsistent closure test

In the following subsection we want to study a more general example than the only-DIS fit previously studied. In this case we are going to show the results for a global inconsistent closure test, thus comprising also hadronic observables. This case is surely more realistic than the previous one studied since it is closer to the standard fitting procedure employed in real data fitting.

Choice of datasets and uncertainties

The inconsistency has been inserted in a Drell-Yan process. In particular we take into consideration the measurement of the double-differential high-mass Drell-Yan cross section in pp collisions at $\sqrt{s} = 8$ TeV measured by the ATLAS collaboration [2].

In this case we are only affecting one dataset, thus the quantity shown in the plot (5.18) is simply:

$$y = \frac{\text{Tr}(C(\lambda))}{\text{Tr}(C_{\text{exp}})}. \quad (5.20)$$

In figure (5.18) we show the evolution of this quantity with $\lambda \rightarrow 0$.

In this case we are actually affecting all the systematic uncertainties which concur in defining the covariance matrix.

Results for inconsistent DY fit

As for the DIS fits we are going to show first the kinematic coverage of the inconsistent data. The number of inconsistent data points in this case is $N_{\text{inc}} = 48$, against a total number of data points for the fitting $N_{\text{fit}} = 3772$: It is clear that in this case we have put an inconsistency in a much smaller number of data points. First of all we are going to show once again the out of sample histograms of the normalized Δ_B . If we take a look only at the histograms of figures (5.20), (5.21) and (5.22), we can see that

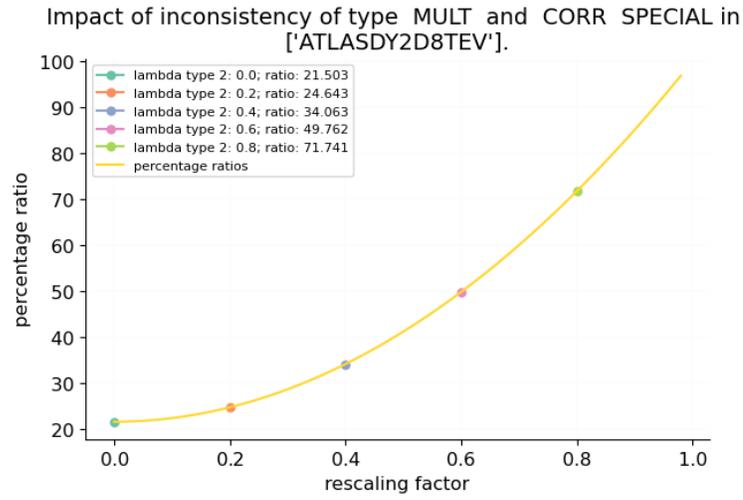


Figure 5.18: Impact on trace of inconsistency

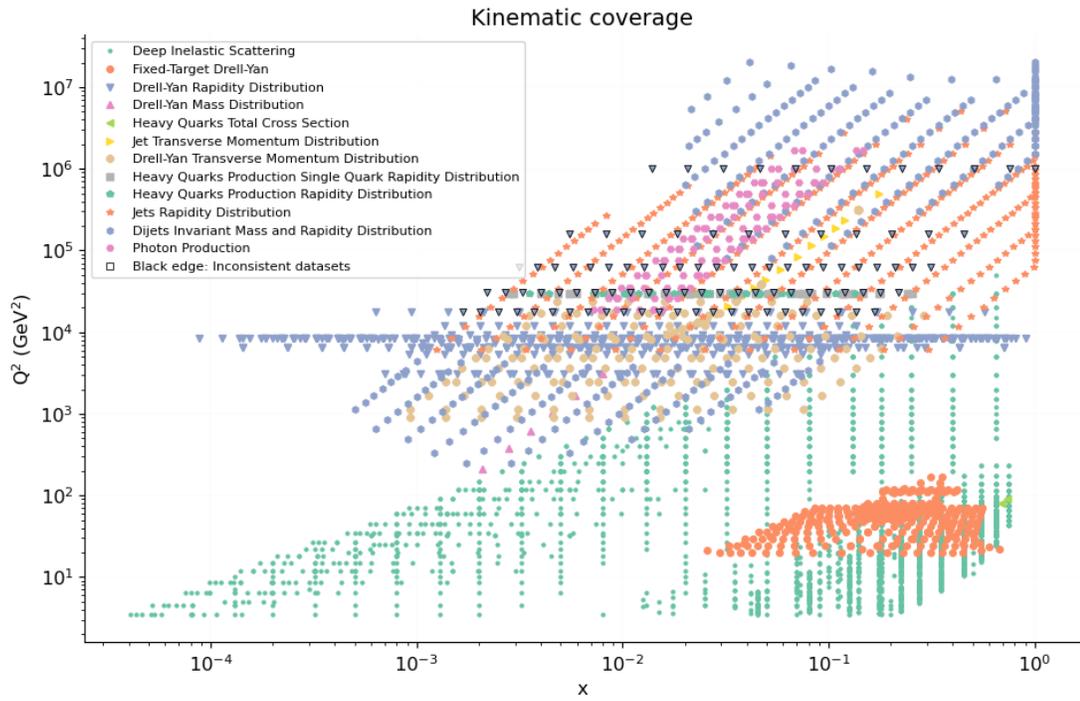
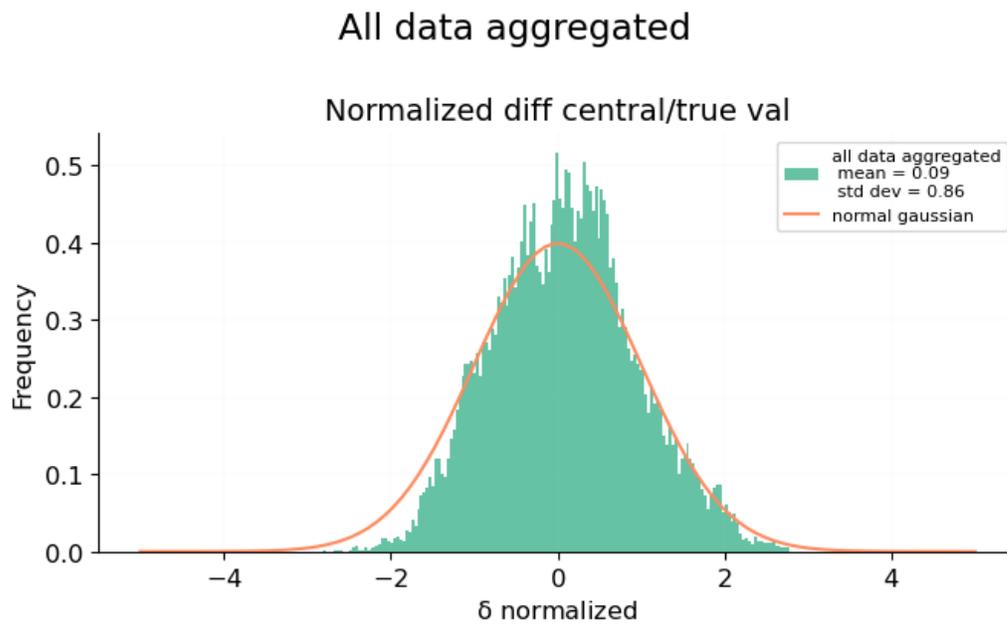
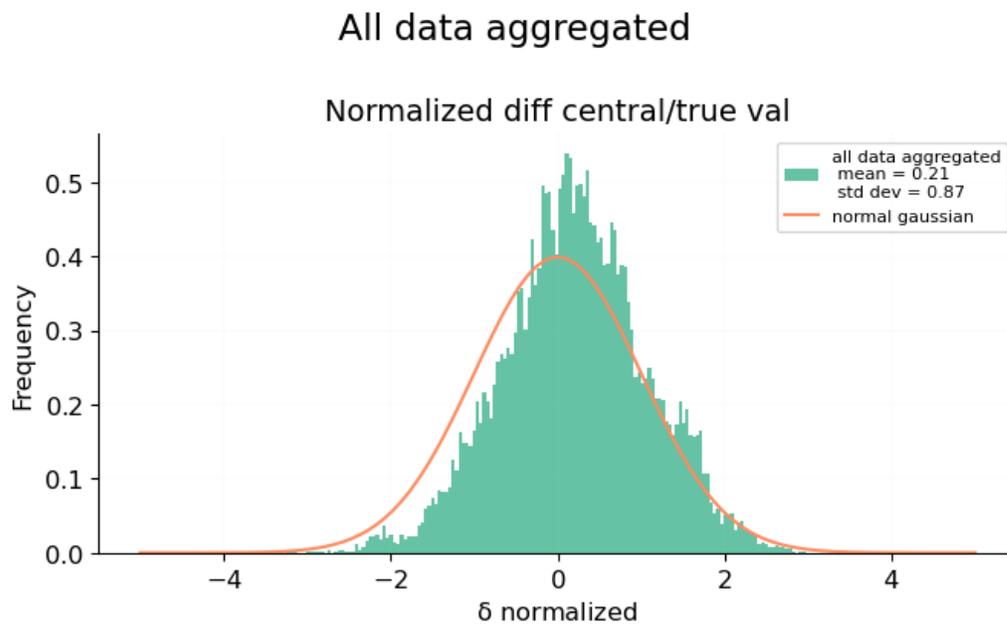


Figure 5.19: Kin coverage inconsistency DY

Figure 5.20: Histogram showing normalized Δ_B for consistent fitFigure 5.21: Histogram showing normalized Δ_B for inconsistent fit $\lambda = 0.8$

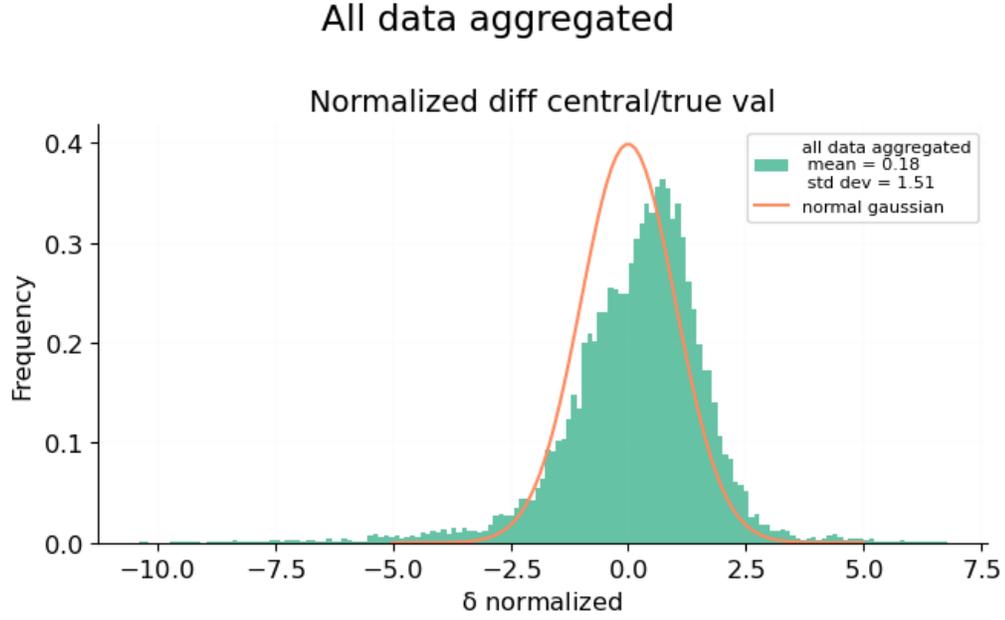


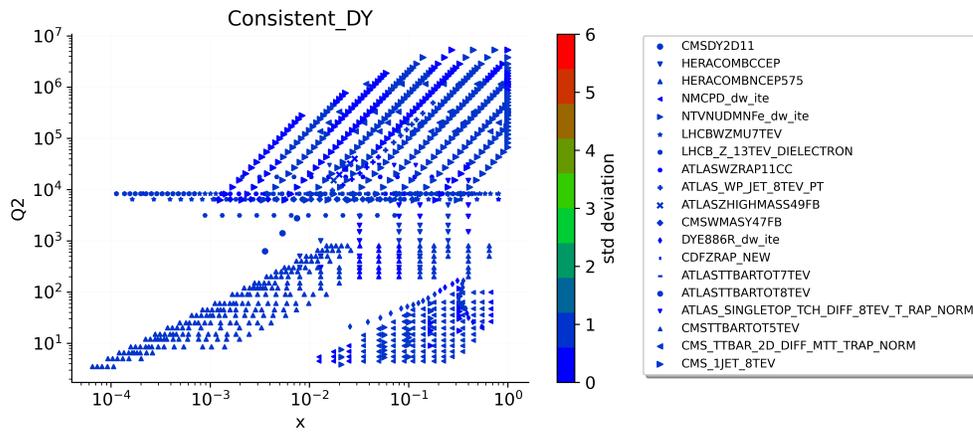
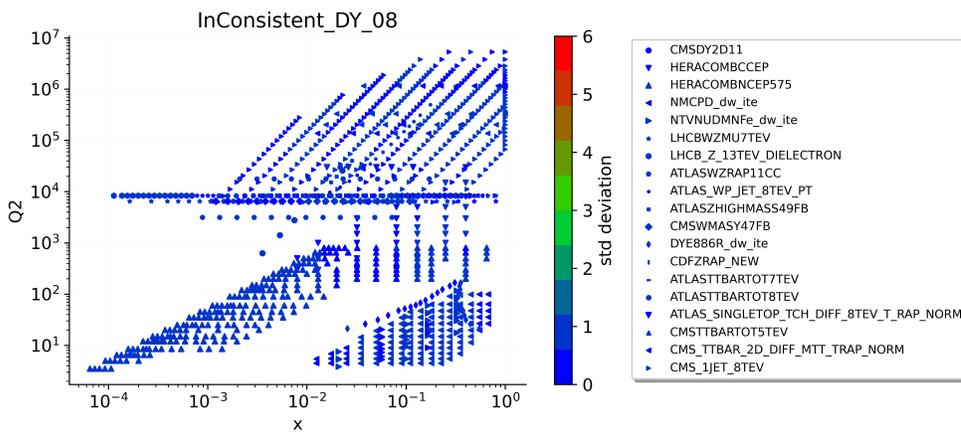
Figure 5.22: Histogram showing normalized Δ_B for inconsistent fit $\lambda = 0.0$

the introduction of the inconsistency yields two effects: on the one hand it broadens the shape of the sample but on the other it also renders it less symmetric. In fact the values of the mean and standard deviation for each fit are:

- Consistent test, $\lambda = 1$: $\mu = 0.09$ and $\sigma = 0.86$,
- Inconsistent test, $\lambda = 0.8$: $\mu = 0.21$ and $\sigma = 0.87$,
- Inconsistent test, $\lambda = 0.0$: $\mu = 0.18$ and $\sigma = 1.51$.

First of all we can see that consistent fit is overestimating uncertainties more than the DIS one, yielding a standard deviation of 0.86. Progressively augmenting the inconsistency we see that in the intermediate case the mean is worsened while the spread of the values remains almost the same. As the previous DIS case, the extreme $\lambda = 0$ inconsistent fit worsens both the mean and the standard deviation by a visible amount.

Viewing the information for the single observables as heatmaps, we can see the broadening effect introduced by the inconsistency in the Drell-Yan observables. Simply by looking at the heat-maps it is safe to say that the impact of the inconsistency is less visible rather than the DIS case. This can be mainly attributed to the size of

Figure 5.23: Standard deviation plot for consistent DIS. $\lambda = 1.0$ Figure 5.24: Standard deviation plot for inconsistent DY. $\lambda = 0.8$

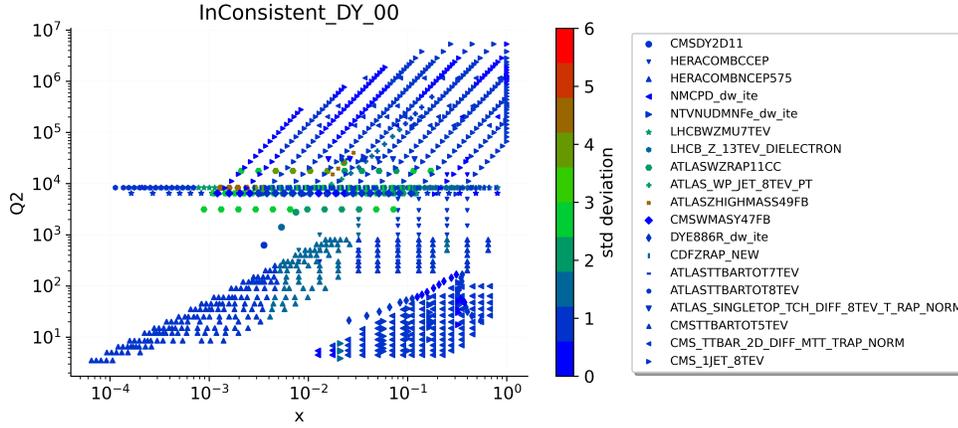


Figure 5.25: Standard deviation plot for inconsistent DY. $\lambda = 0.0$

the inconsistent dataset; anyway also in this case there is a correspondence between introduced inconsistency and out of sample performance. This correspondence can be seen more clearly in the histograms in the appendix, in which it will be easier to see which are the impacted data points.

Conclusions for Drell Yan fits

The inconsistent datasets in this case covered a much smaller portion in data space compared to the DIS case previously studied. It is interesting to notice that in this case the change in the global standard deviation for the intermediate inconsistency case is even less visible when compared to the DIS one; on the other hand we can see a visible worsening of the mean of the normalized Δ_B . This phenomenon might be related to two things: first of all, as already said, performing the analysis in data space implies having to deal with correlated variables which might bias the result. On the other hand this could be related to the presence of hadronic observable in the global set, which might introduce sources of non-gaussianity in the distribution of the normalized Δ_B s.

As in the case of DIS observables the change in the standard deviation becomes really visible only in the extreme $\lambda = 0$ case. It is interesting to notice that here the ratio of inconsistent data to whole training size is much smaller than the DIS case, but we are still able to see a worsening of the NN performance overall.

Jets inconsistencies

In the following subsection we are going to show the setup and results for inconsistencies in jets cross section data. The chosen data are still measured by the ATLAS collaboration and they refer to the measurement of the inclusive jet cross-sections in proton-proton collisions at $\sqrt{s} = 8$ TeV [1]. Once again we show the weight of the uncertainties which were made inconsistent in this study. Also here we take into

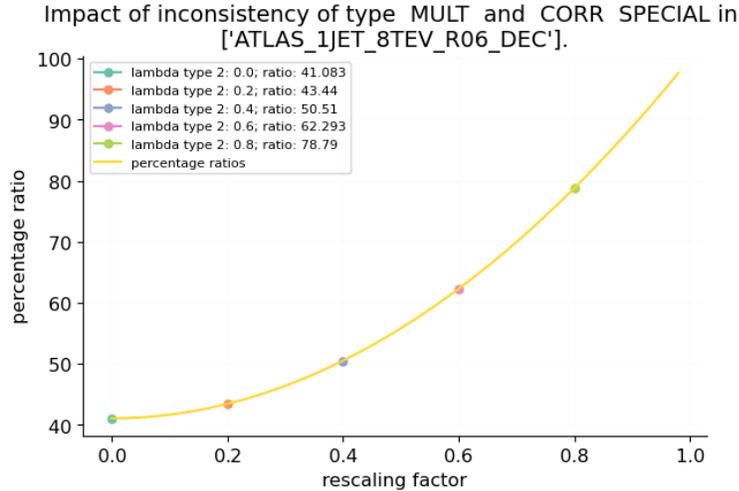


Figure 5.26: Impact on trace of inconsistency

consideration an extreme example, affecting all the correlated uncertainties in the chosen dataset.

Results for inconsistent JETS fit

As in the previous examples we want to highlight the trend with λ of the histograms plotting the normalized Δ_B s. First of all we show the kinematic coverage of the inconsistent data with respect to the whole training set. In this case the number of inconsistent data points is $N_{\text{inc}} = 171$ against a total of training points. $N_{\text{tr}} = 3793$. We show again the histograms for the normalized Δ_B for the whole out of sample testing set. Also In this case it can be seen that the overall fit quality is worsened by introduction of inconsistencies. In particular the mean and standard deviation of the fits take the following values:

- Consistent test, $\lambda = 1$: $\mu = 0.17$ and $\sigma = 0.84$,

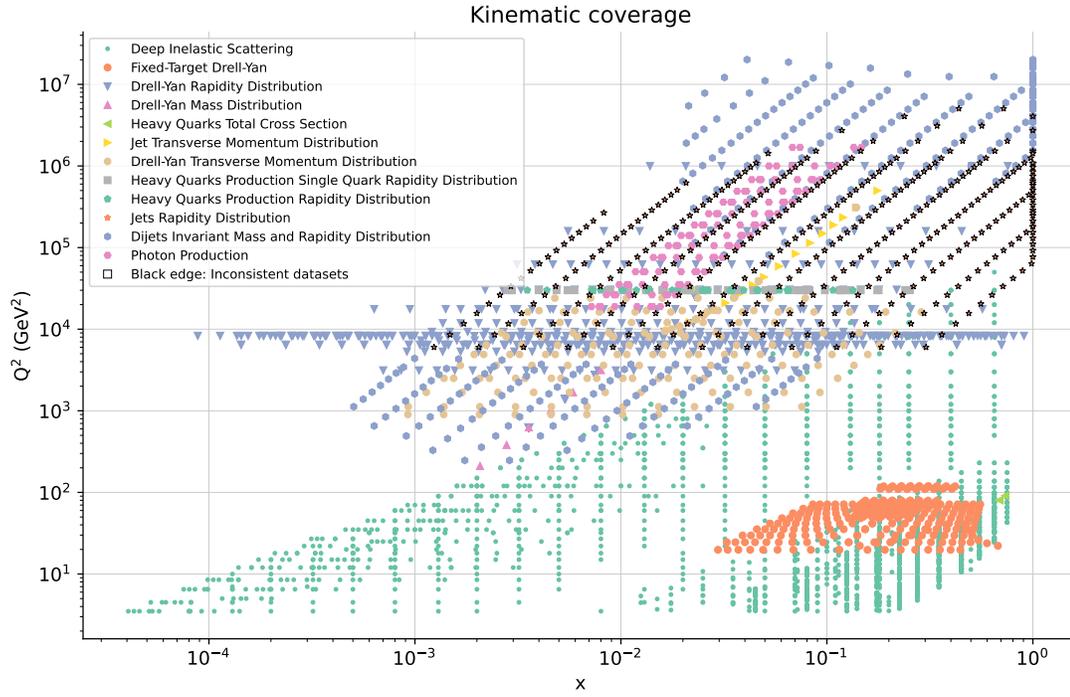
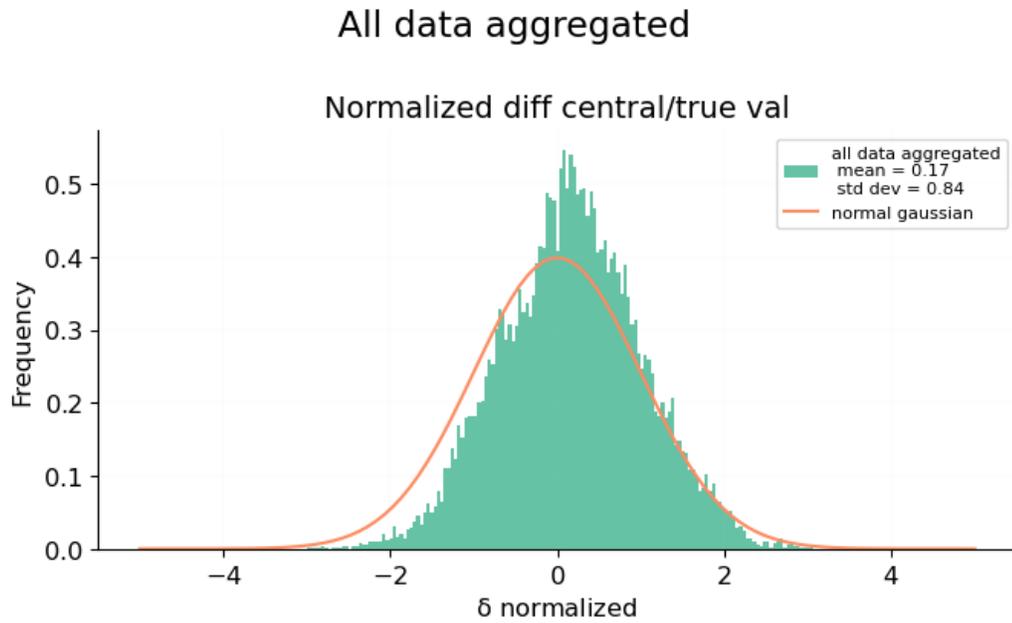


Figure 5.27: Kin coverage inconsistency JETS

Figure 5.28: Histogram showing normalized Δ_B for consistent fit

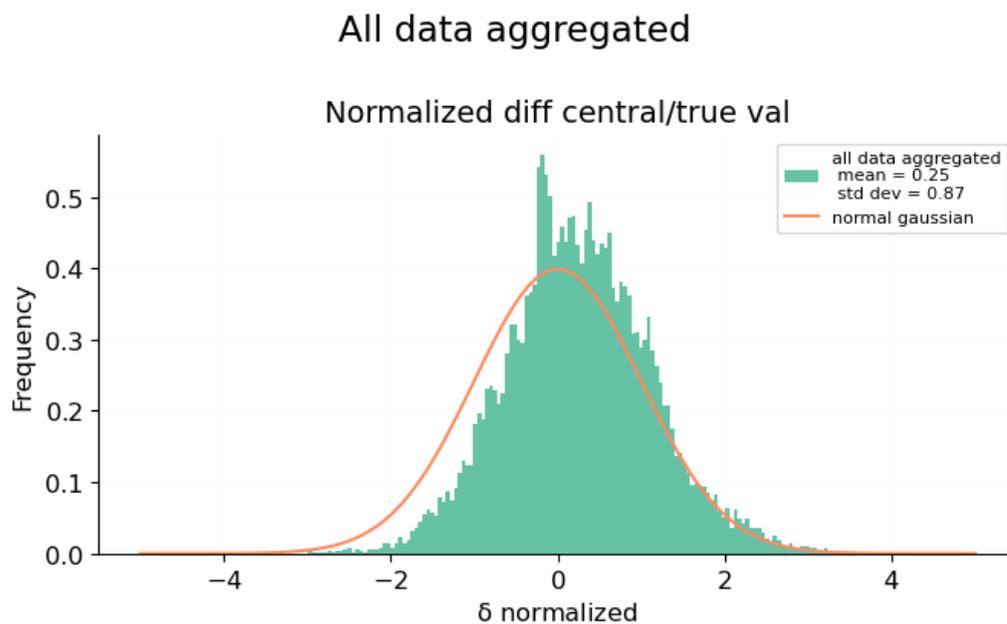


Figure 5.29: Histogram showing normalized Δ_B for inconsistent fit $\lambda = 0.6$

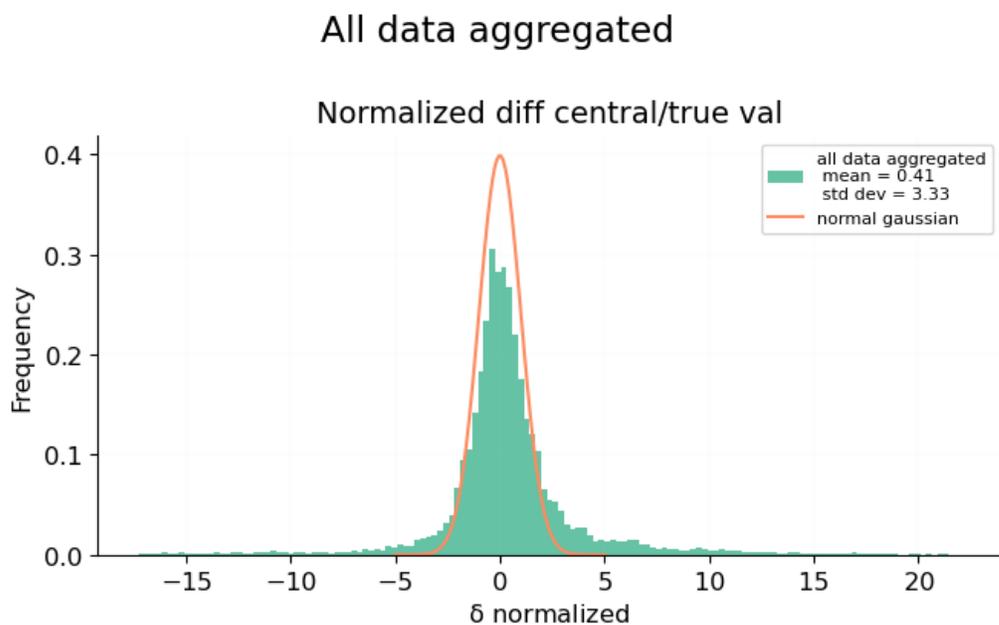


Figure 5.30: Histogram showing normalized Δ_B for inconsistent fit $\lambda = 0.0$

- Inconsistent test, $\lambda = 0.6$: $\mu = 0.25$ and $\sigma = 0.87$,
- Inconsistent test, $\lambda = 0.0$: $\mu = 0.41$ and $\sigma = 3.33$.

Also in this case we see a trend which is similar to the Drell Yan case: in particular also here we see a corruption in the mean visible in the middle point of the inconsistency ($\lambda = 0.6$) while the standard deviation remains almost the same. On the other hand in this case the most inconsistent case shows an even larger increase in the standard deviation: this can be safely attributed to the fact that the size of the points which were made inconsistent is much larger than before, roughly 4 times more.

In order to check the single data point performance of the NN we also show the heatmap of the predictions:

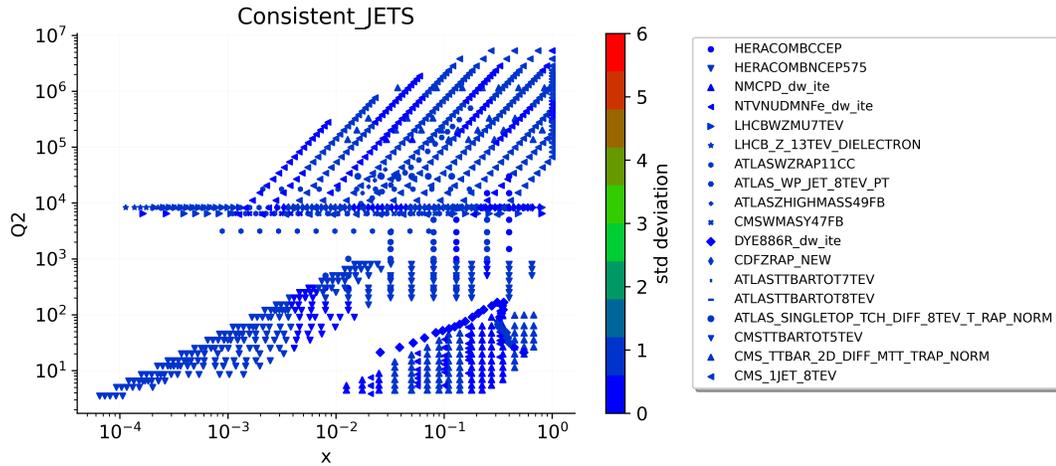
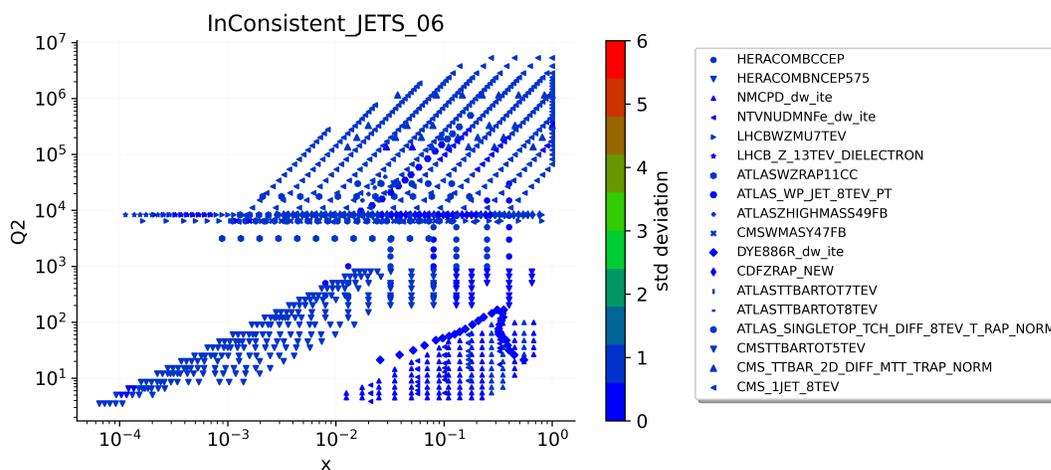
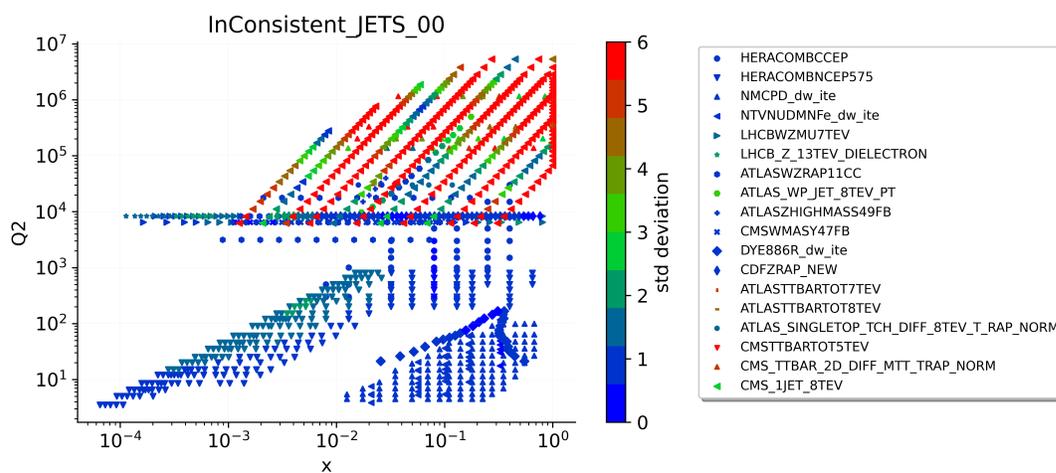


Figure 5.31: Standard deviation plot for consistent JETS. $\lambda = 1.0$

In this case it is evident that the region over which the NN performs badly is much broader than the Drell Yan case, once again given the larger size of inconsistent data in the training. It is still interesting to notice that a correspondence between inconsistent observables in the training and badly behaving ones in the out of sample testing does not cease to hold.

Conclusions for JETS fits

In this second case of global inconsistent fit, we notice a trend similar to the DY case. The main difference with the DY case is that we inserted the inconsistency

Figure 5.32: Standard deviation plot for inconsistent JETS. $\lambda = 0.6$ Figure 5.33: Standard deviation plot for inconsistent JETS. $\lambda = 0.0$

in a much broader set of data points. This can be easily seen by the fact that the overall standard deviation of the normalized Δ_B is much larger for the $\lambda = 0$ case and consequently that many more observables perform badly.

It is interesting to notice that also here the mean of the R.V. Δ_B is shifted from the expected value 0: this once again suggests that the presence of hadronic observables affects the behaviour of this figure of merit. Finally we can see also here that the NN learns the inconsistency in the extreme case propagating it essentially to the same kinds of observable.

5.3.1 Final remarks

In this conclusive section we want to sum up the findings of this study.

We have inserted an inconsistency in three kinds of different setups: one only-DIS setup and two global fits, making respectively Drell-Yan and Jets observables inconsistent. We have studied extreme examples since we corrupted a large part of the systematic uncertainties affecting particular datasets. In all three setups we varied the size of the introduced inconsistency by varying a parameter λ ranging from 1 to 0.

All the three cases of study show common characteristics: in the intermediate situation with $\lambda \in (0, 1)$ the NN is able to reabsorb the inconsistency and it performs as if no inconsistency was introduced. On the other hand in the extreme case of $\lambda = 0$ in all three cases the performance of the NN is heavily worsened, overall overestimating uncertainties.

Another feature common to all three setups is the fact that the NN propagates inconsistency in data space only in those regions which have the same characteristics as the inconsistent datasets. In all three setups, given an inconsistent process in the training data, the same process in the out of sample testing set shows the worst performance of the NN.

A feature which is peculiar only to the two global fits is the fact that in both the mean of the normalized Δ_B is greatly shifted away from 0 as the inconsistency increases. This could be caused by the non-linearity in the PDFs of the observables which were made inconsistent during the training.

There are a few possible follow-ups to this work: first of all the problem to be properly addressed is the correlation in data space induced by the forward map. In this work we relied on the homogeneity of the testing set which can justify the results

of this work. A possible way of solving this problem would be to actually build from scratch a testing set which is free of problems when dealing with uncertainties.

Conclusions

At the beginning of this work the idea was to expand previous studies related to the inconsistent closure tests, which can be found in [20]. Specifically, our aim was to conduct more realistic inconsistent closure tests than those eventually executed, involving the introduction of significantly greater inconsistencies.

Throughout our research, we discovered the necessity to revise the closure test formalism, prompting us to make adjustments to the previous approach.

In the first part of this work, we changed the figure of merit used as a diagnostic for the NN performance, partially removing the bias which affected the previously adopted one. Employing this new methodology, we can say that the standard functioning of the NN slightly overestimates uncertainties: still for a perfect assessment of the NN performance, additional research in this direction is required.

The second part of the work is devoted to the inconsistent closure tests. The first inconsistent closure test has been performed in a simple setting: we included only DIS observables which at LO are linear in the PDFs. The two other cases include also hadronic observables, and the inconsistency has been placed first in a set including Drell Yan process data, and in second place in a set of Jet cross sections data.

The studied examples are extreme in the sense that we are simulating the situation in which almost all the uncertainties were estimated in the wrong way. This is for sure more unrealistic if compared to previous studies regarding inconsistent closure test, but the good advantage is that it can give us more insights into the response of the NN to inconsistencies. In fact it is interesting to notice that also in this extreme case the NN performs in a consistent way if the systematic uncertainties are only rescaled by a λ factor below 1 but different from zero; only in the case in which we completely remove the systematic uncertainties the NN starts performing in an inconsistent way.

In second place we can see that in the extreme case in which we set $\lambda = 0$ the NN affects only the testing data which measure the same observables made inconsistent

during the training. This suggests that the NN somehow is able to distinguish the various kind of processes, translating the inconsistency in data space to the PDF level following a simple pattern.

One crucial aspect requiring further attention in future studies is the rigorous handling of correlations during the evaluation of the NN in data space. Although this thesis has made strides in enhancing previous results, it still exhibits a slight bias when assessing the performance of the NN in data space. Future research in this area holds significant importance as it enables a more nuanced examination of inconsistent closure tests in increasingly realistic scenarios, where the magnitude of inconsistency is reduced compared to the cases under investigation.

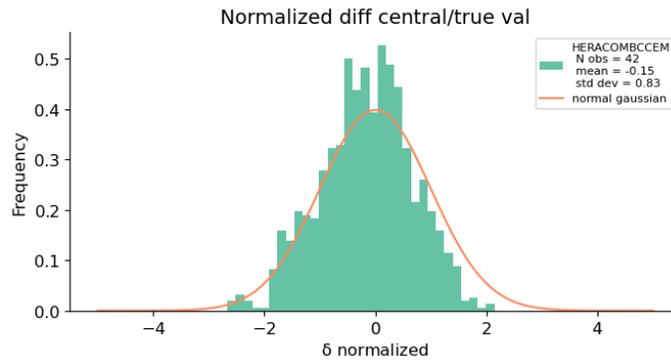
Appendix A

Further results

In the following appendix we are going to list a few more histograms which have not been shown in the main body of the work for presentation reasons.

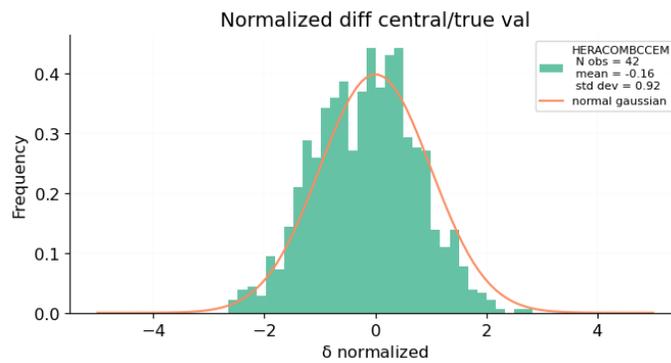
A.1 DIS fits

HERACOMBCCEM; N fits = 25



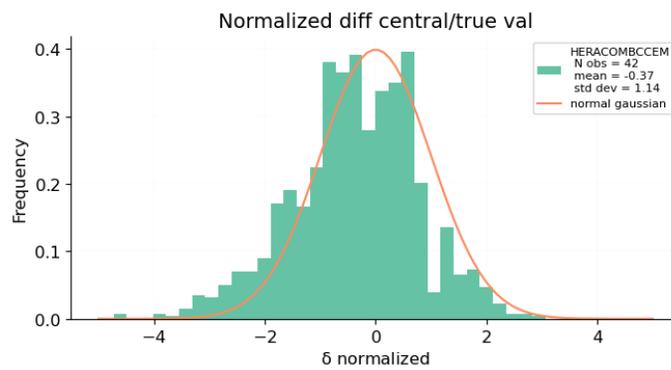
(a) $\lambda = 1$; DIS consistent fit

HERACOMBCCEM; N fits = 26



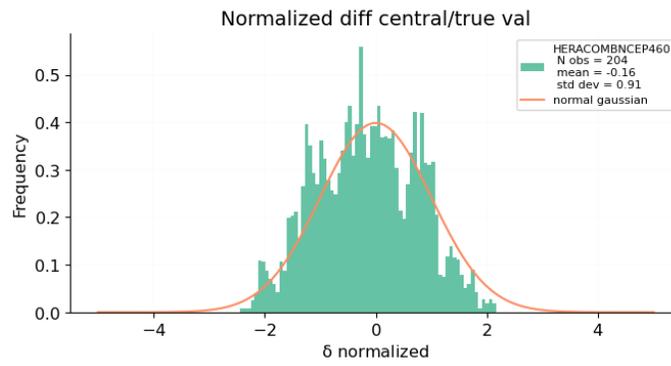
(b) $\lambda = 0.6$; DIS inconsistent fit

HERACOMBCCEM; N fits = 26

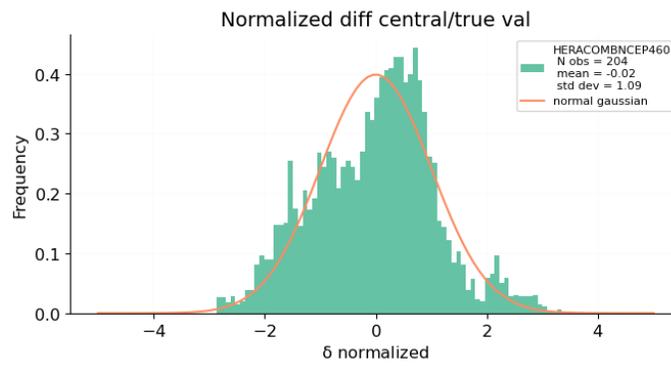


(c) $\lambda = 0.0$; DIS inconsistent fit

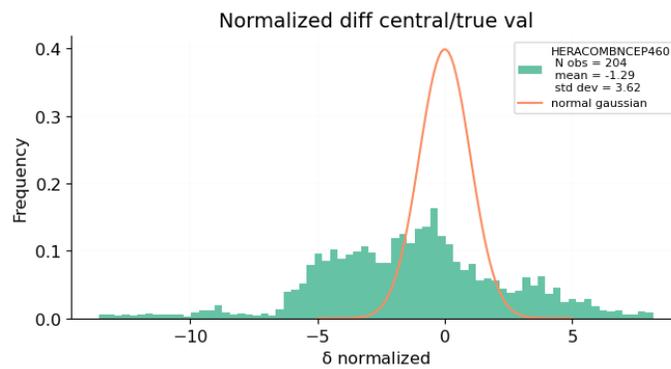
HERACOMBNCEP460; N fits = 25

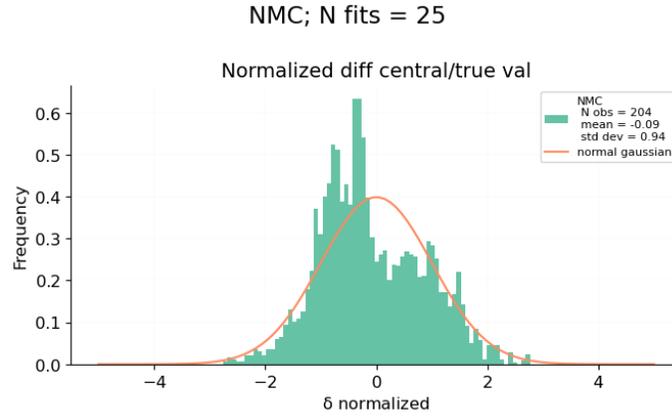
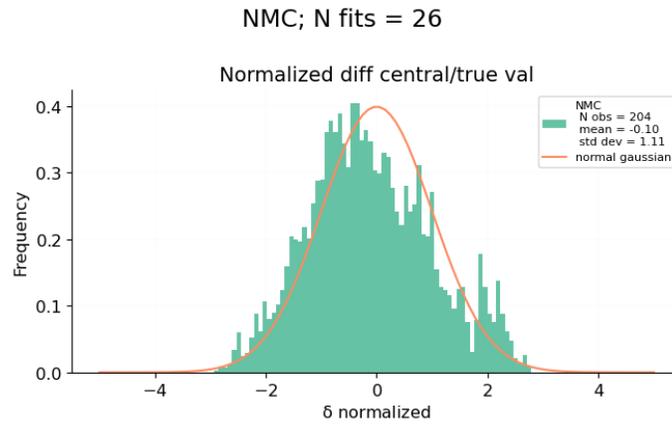
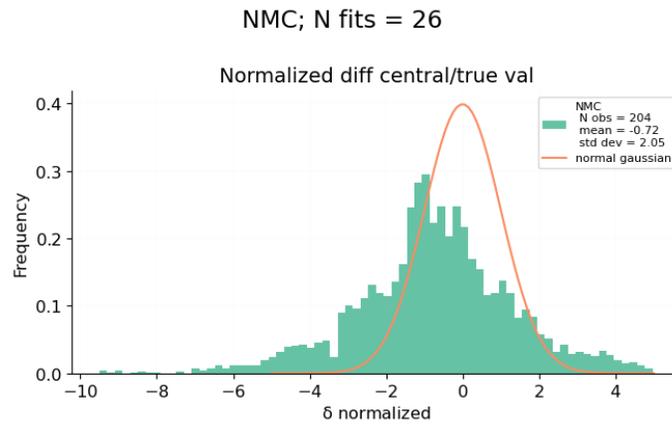
(a) $\lambda = 1$; DIS consistent fit

HERACOMBNCEP460; N fits = 26

(b) $\lambda = 0.6$; DIS inconsistent fit

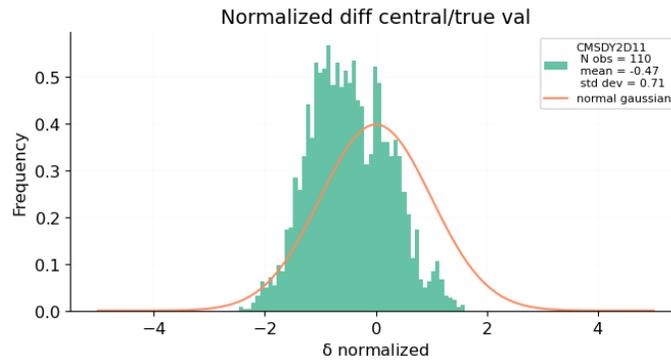
HERACOMBNCEP460; N fits = 26

(c) $\lambda = 0.0$; DIS inconsistent fit

(a) $\lambda = 1$; DIS consistent fit(b) $\lambda = 0.6$; DIS inconsistent fit(c) $\lambda = 0.0$; DIS inconsistent fit

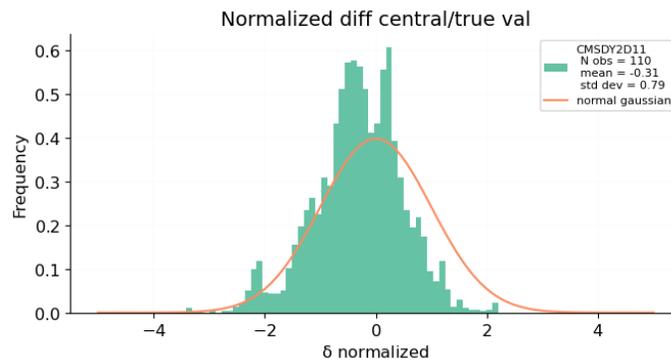
A.2 DY fits

CMSDY2D11; N fits = 27



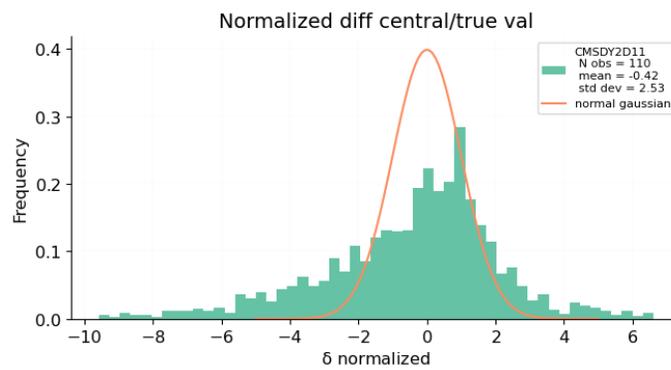
(a) $\lambda = 1$; DY consistent fit

CMSDY2D11; N fits = 26



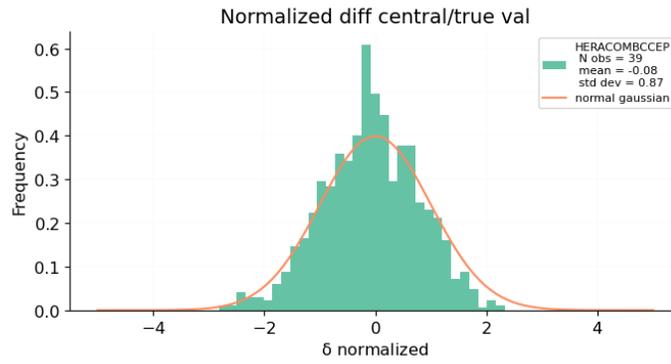
(b) $\lambda = 0.8$; DY inconsistent fit

CMSDY2D11; N fits = 26

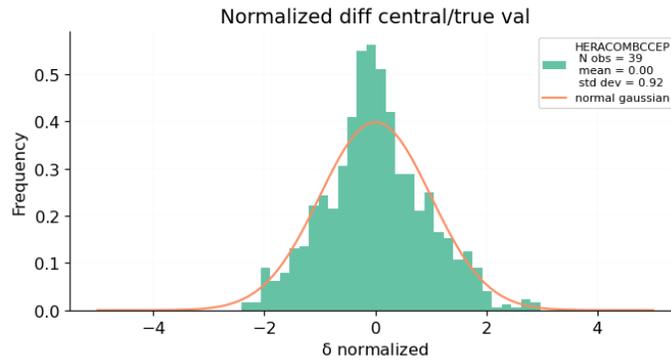


(c) $\lambda = 0.0$; DY inconsistent fit

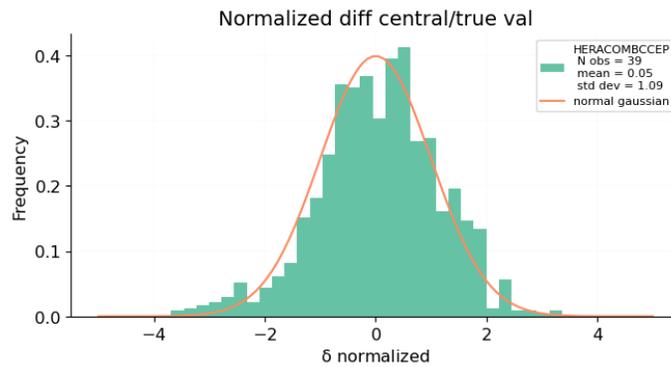
HERACOMBCCEP; N fits = 27

(a) $\lambda = 1$; DY consistent fit

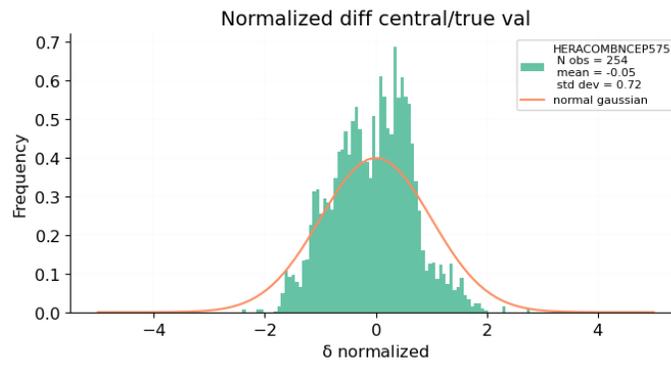
HERACOMBCCEP; N fits = 26

(b) $\lambda = 0.8$; DY inconsistent fit

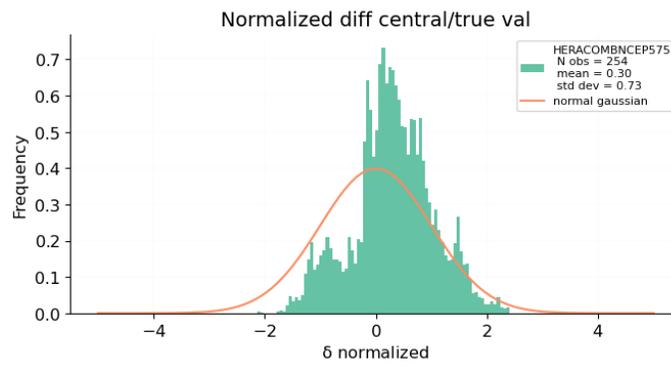
HERACOMBCCEP; N fits = 26

(c) $\lambda = 0.0$; DY inconsistent fit

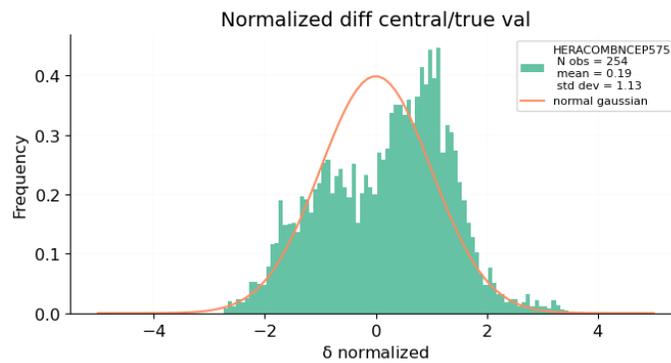
HERACOMBNCEP575; N fits = 27

(a) $\lambda = 1$; DY consistent fit

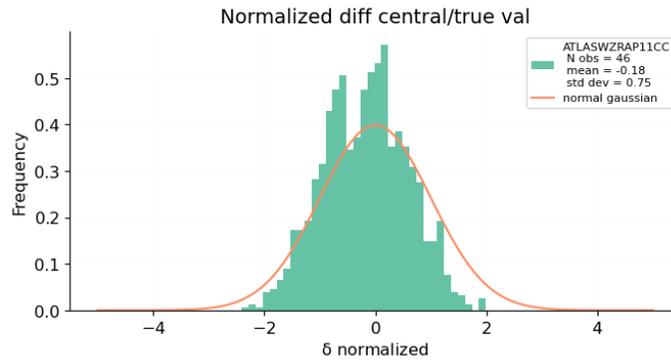
HERACOMBNCEP575; N fits = 26

(b) $\lambda = 0.8$; DY inconsistent fit

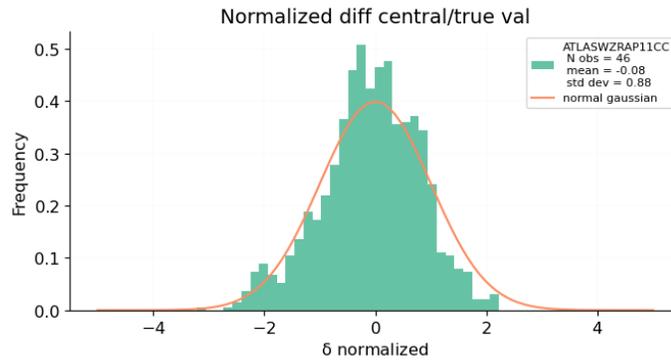
HERACOMBNCEP575; N fits = 26

(c) $\lambda = 0.0$; DY inconsistent fit

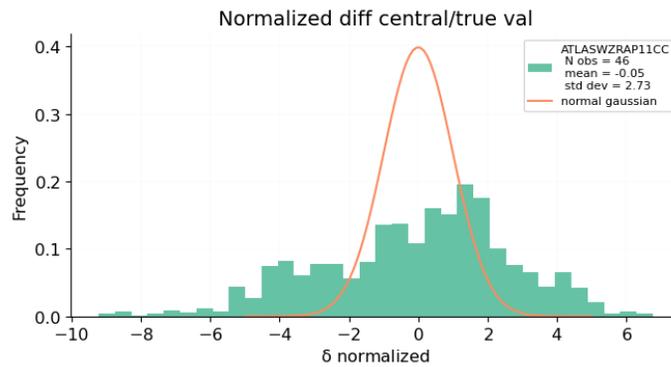
ATLASWZRAP11CC; N fits = 27

(a) $\lambda = 1$; DY consistent fit

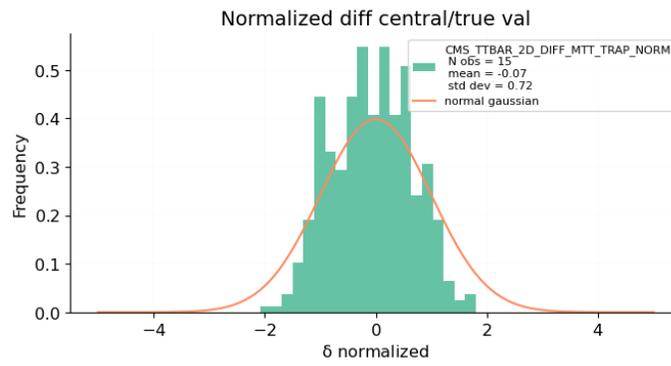
ATLASWZRAP11CC; N fits = 26

(b) $\lambda = 0.8$; DY inconsistent fit

ATLASWZRAP11CC; N fits = 26

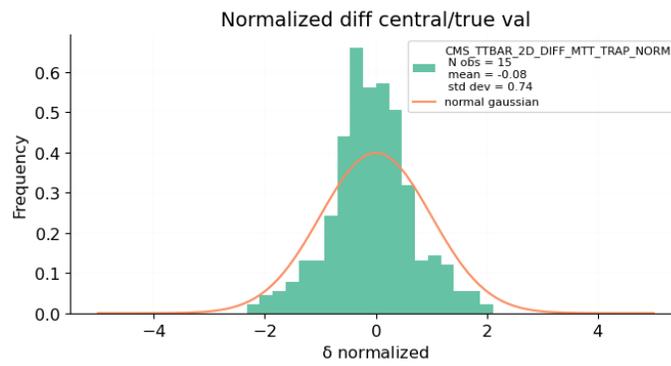
(c) $\lambda = 0.0$; DY inconsistent fit

CMS_TTBAR_2D_DIFF_MTT_TRAP_NORM; N fits = 27



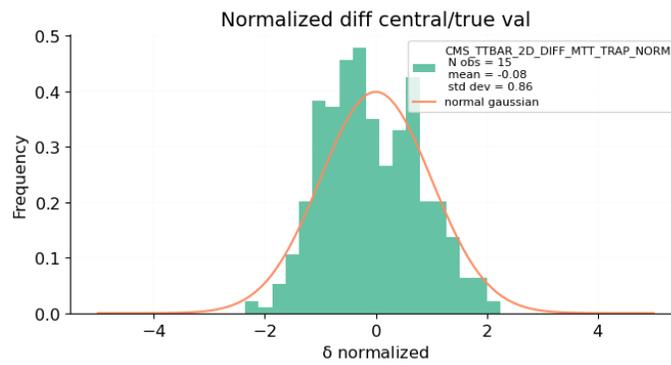
(a) $\lambda = 1$; DY consistent fit

CMS_TTBAR_2D_DIFF_MTT_TRAP_NORM; N fits = 26



(b) $\lambda = 0.8$; DY inconsistent fit

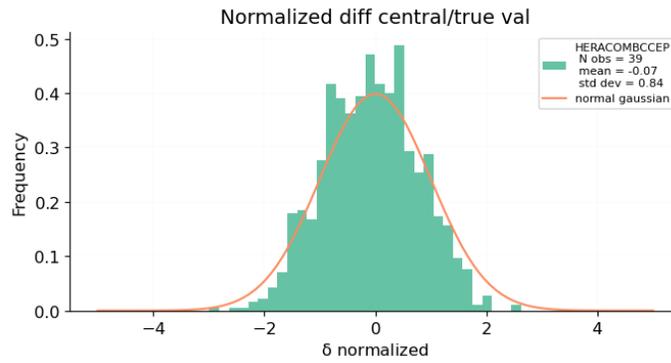
CMS_TTBAR_2D_DIFF_MTT_TRAP_NORM; N fits = 26



(c) $\lambda = 0.0$; DY inconsistent fit

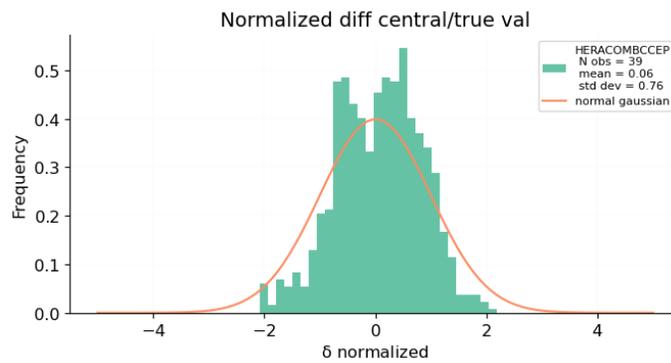
A.3 JETS fits

HERACOMBCCEP; N fits = 27



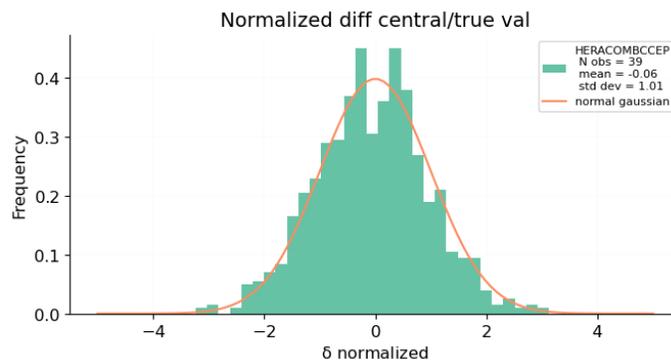
(a) $\lambda = 1$; JETS consistent fit

HERACOMBCCEP; N fits = 23



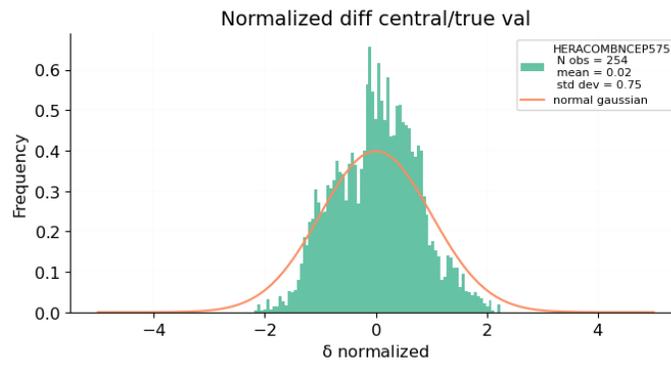
(b) $\lambda = 0.6$; JETS inconsistent fit

HERACOMBCCEP; N fits = 25

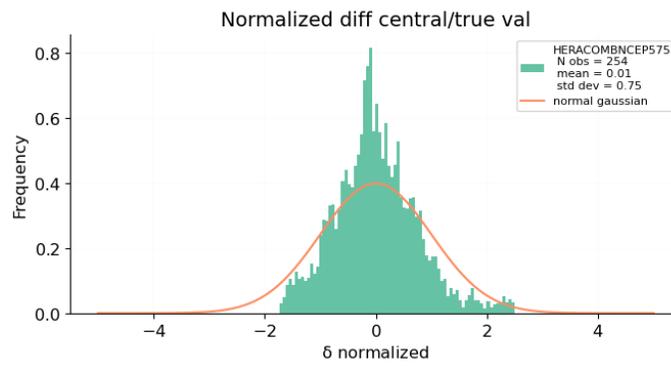


(c) $\lambda = 0.0$; JETS inconsistent fit

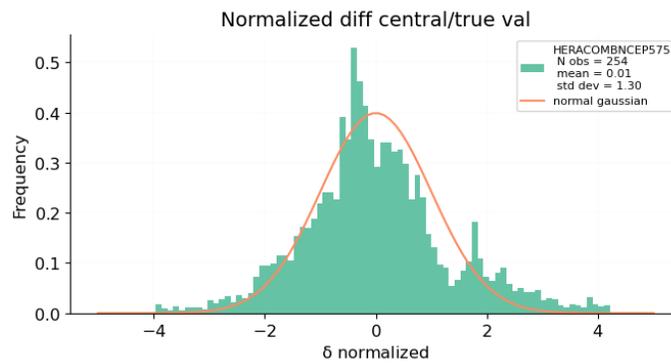
HERACOMBNCEP575; N fits = 27

(a) $\lambda = 1$; JETS consistent fit

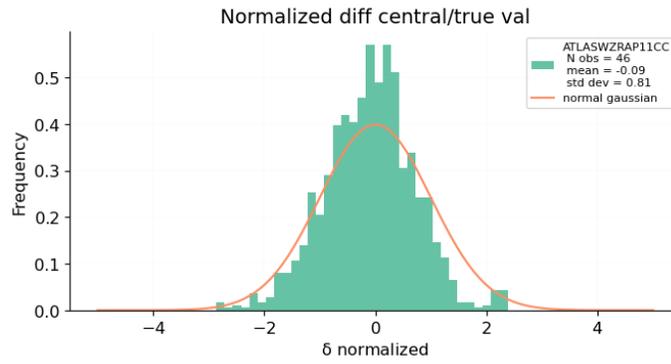
HERACOMBNCEP575; N fits = 23

(b) $\lambda = 0.6$; JETS inconsistent fit

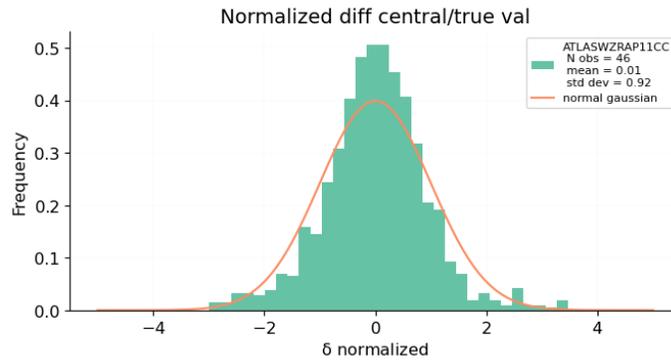
HERACOMBNCEP575; N fits = 25

(c) $\lambda = 0.0$; JETS inconsistent fit

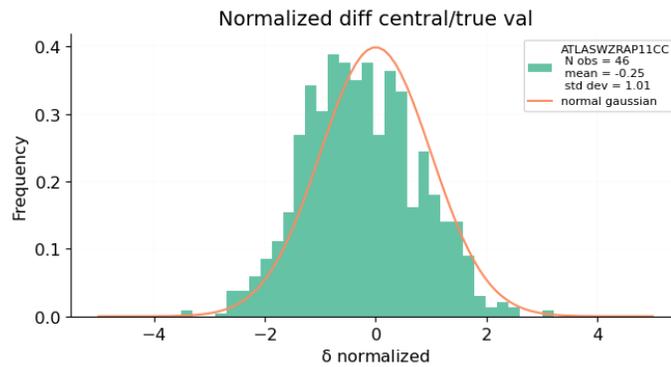
ATLASWZRAP11CC; N fits = 27

(a) $\lambda = 1$; JETS consistent fit

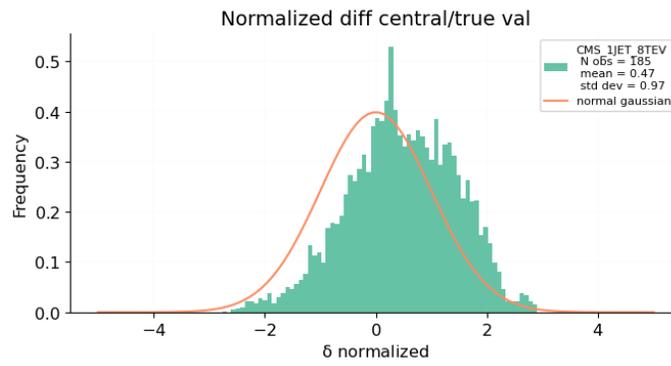
ATLASWZRAP11CC; N fits = 23

(b) $\lambda = 0.6$; JETS inconsistent fit

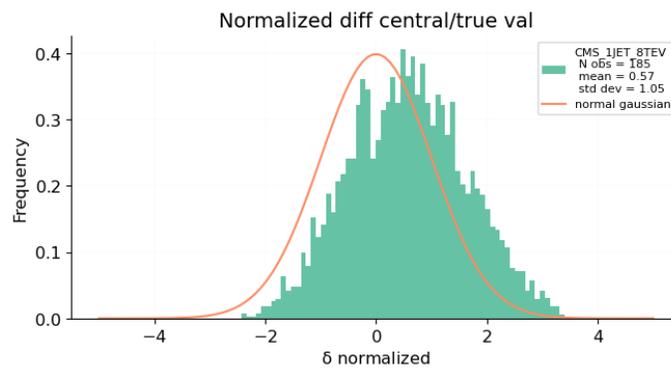
ATLASWZRAP11CC; N fits = 25

(c) $\lambda = 0.0$; JETS inconsistent fit

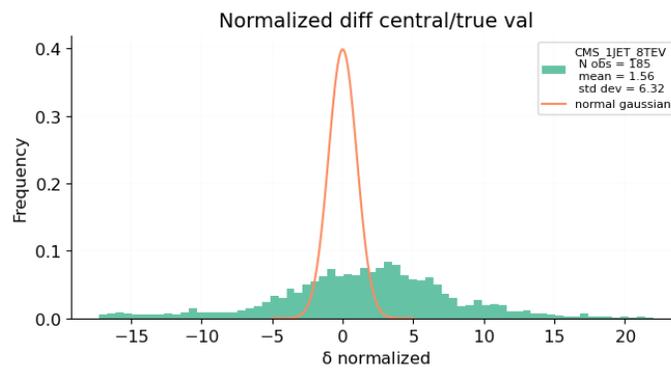
CMS_1JET_8TEV; N fits = 27

(a) $\lambda = 1$; JETS consistent fit

CMS_1JET_8TEV; N fits = 23

(b) $\lambda = 0.6$; JETS inconsistent fit

CMS_1JET_8TEV; N fits = 25

(c) $\lambda = 0.0$; JETS inconsistent fit

Bibliography

- [1] Morad Aaboud et al. “Measurement of the inclusive jet cross-sections in proton-proton collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector”. In: *J. High Energy Phys.* 9 (2017), pp. 1–54.
- [2] Georges Aad et al. “Measurement of the double-differential high-mass Drell-Yan cross section in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector”. In: *J. High Energy Phys.* 8 (2016), pp. 1–61.
- [3] Francise D. Aaron et al. “Combined measurement and QCD analysis of the inclusive $e^\pm p$ scattering cross sections at HERA”. In: *J. High Energy Phys.* 1 (2010), pp. 1–63.
- [4] Halina Abramowicz et al. “Combination of measurements of inclusive deep inelastic $e^\pm p$ scattering cross sections and QCD analysis of HERA data: H1 and ZEUS Collaborations”. In: *Eur. Phys. J. C* 75 (2015), pp. 1–98.
- [5] J. R. Andersen et al. “Les Houches 2015: Physics at TeV Colliders Standard Model Working Group Report”. In: *9th Les Houches Workshop on Physics at TeV Colliders*. May 2016. arXiv: 1605.04692 [hep-ph].
- [6] Richard D. Ball et al. “Parton distributions for the LHC Run II”. In: *J. High Energy Phys.* 4 (2015), pp. 1–148.
- [7] Richard D. Ball et al. “The path to proton structure at 1% accuracy: NNPDF Collaboration”. In: *Eur. Phys. J. C* 82.5 (2022), p. 428.
- [8] Andy Buckley et al. “LHAPDF6: parton density access in the LHC precision era”. In: *Eur. Phys. J. C* 75 (2015), pp. 1–20.
- [9] Alessandro Candido, Stefano Forte, and Felix Hekhorn. “Can MS parton distributions be negative?” In: *J. High Energy Phys.* 11 (2020), pp. 1–30.

- [10] Stefano Carrazza and Juan Cruz-Martinez. “Towards a new generation of parton densities with deep learning models”. In: *Eur. Phys. J. C* 79 (2019), pp. 1–9.
- [11] Stefano Carrazza, Juan Cruz-Martinez, and Roy Stegeman. “A data-based parametrization of parton distribution functions”. In: *Eur. Phys. J. C* 82.2 (2022), p. 163.
- [12] Stefano Carrazza et al. “An unbiased Hessian representation for Monte Carlo PDFs”. In: *Eur. Phys. J. C* 75 (2015), pp. 1–20.
- [13] Luigi Del Debbio, Tommaso Giani, and Michael Wilson. “Bayesian approach to inverse problems: an application to NNPDF closure testing”. In: *Eur. Phys. J. C* 82.4 (2022), p. 330.
- [14] Luigi Del Debbio et al. “Unbiased determination of the proton structure function F_2^p with faithful uncertainty estimation”. In: *J. High Energy Phys.* 3 (2005), p. 80.
- [15] R. Keith Ellis, W. James Stirling, and Bryan R. Webber. *QCD and Collider Physics*. Cambridge Univ. Press, 2010.
- [16] Stefano Forte and Stefano Carrazza. “Parton distribution functions”. In: *Artificial Intelligence For High Energy Physics*. World Scientific, 2022, pp. 715–762.
- [17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366.
- [18] Michael E. Peskin. *An Introduction to Quantum Field Theory*. Addison-Wesley Publishing Company, Advanced Book Program, Reading, MA, 1995. CRC press, 1995.
- [19] Andrew M. Stuart. “Inverse problems: a Bayesian perspective”. In: *Acta Numer.* 19 (2010), pp. 451–559.
- [20] Samuele Voltan. *Validation criteria in the determination of parton distributions*. Master’s thesis. Available at <https://n3pdf.mi.infn.it/documents/theses/>. 2022.