



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Neural network determination of parton distributions

Citation for published version:

NNPDF Collaboration, T, Del Debbio, L, Forte, S, Iatorre, J, Piccione, A & Rojo, J 2007, 'Neural network determination of parton distributions: the nonsinglet case', *Journal of High Energy Physics*, vol. 2007, no. 039, pp. -. <https://doi.org/10.1088/1126-6708/2007/03/039>

Digital Object Identifier (DOI):

[10.1088/1126-6708/2007/03/039](https://doi.org/10.1088/1126-6708/2007/03/039)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Published In:

Journal of High Energy Physics

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Neural network determination of parton distributions: the nonsinglet case

The NNPDF Collaboration:

Luigi Del Debbio¹, Stefano Forte², José I. Latorre³, Andrea Piccione⁴ and Joan Rojo^{3,5}

¹ *SUPA, School of Physics, University of Edinburgh, Edinburgh EH9 3JZ, Scotland*

² *Dipartimento di Fisica, Università di Milano and*

INFN, Sezione di Milano, Via Celoria 16, I-20133 Milano, Italy

³ *Departament d'Estructura i Constituents de la Matèria,*

Universitat de Barcelona, Diagonal 647, E-08028 Barcelona, Spain

⁴ *INFN, Sezione di Genova, via Dodecaneso 33, I-16146 Genova, Italy*

⁵ *LPTHE, CNRS UMR 7589*

Universités Paris VI-Paris VII, F-75252, Paris Cedex 05, France

ABSTRACT: We provide a determination of the isotriplet quark distribution from available deep-inelastic data using neural networks. We give a general introduction to the neural network approach to parton distributions, which provides a solution to the problem of constructing a faithful and unbiased probability distribution of parton densities based on available experimental information. We discuss in detail the techniques which are necessary in order to construct a Monte Carlo representation of the data, to construct and evolve neural parton distributions, and to train them in such a way that the correct statistical features of the data are reproduced. We present the results of the application of this method to the determination of the nonsinglet quark distribution up to next-to-next-to-leading order, and compare them with those obtained using other approaches.

KEYWORDS: QCD, neural networks, parton distributions.

Contents

1. Introduction	2
2. General strategy	4
3. Experimental data and Monte Carlo generation	9
3.1 Experimental data and kinematical cuts	9
3.2 Monte Carlo generation	10
4. From parton distributions to physical observables	11
4.1 Leading-twist evolution and coefficient functions	12
4.2 Solution of the evolution equation in N space	13
4.3 Evolution kernel in x -space	15
4.4 Target mass corrections and higher twists	18
5. Neural networks	22
5.1 Structure	22
5.2 Training	24
5.3 Stopping	26
6. Results	28
6.1 Next-to-leading order results: central values	28
6.2 Next-to-leading order results: uncertainties	30
6.3 Stability upon variation of the neural network structure	32
7. Theoretical uncertainties	33
7.1 Dependence on kinematic cuts	34
7.2 Higher twists	34
7.3 Higher orders and the value of the strong coupling	35
8. The structure function	37
8.1 Comparison to data and other parton fits	37
8.2 Comparison to neural $F_2^{\text{NS}}(x, Q^2)$	39
9. Conclusions and outlook	39
A. Large x behavior of the evolution factor	41
B. Statistical estimators	41

1. Introduction

The needs of precision physics at hadron colliders have determined a revolution in the approach to the determination of parton distributions functions (PDFs) over the last few years. While the Tevatron has been providing data for a variety of hard hadronic processes which establish the validity and accuracy of perturbative factorization and parton universality to a level which is now comparable to that of precision tests of the standard model in the electroweak sector, LHC, now behind the corner, will require, essentially for the first time, a precision approach to the structure of the nucleon in the context of searches for new physics [1]. Over a decade of experimental effort, especially in deep-inelastic experiments, first and foremost at the HERA collider [2], but also from neutrino beams and with muon beams on a fixed target, has provided us with an unprecedented amount of information which makes such a precision approach possible. However, we have now reached a stage in which the development of new theoretical and phenomenological analysis tools is needed [3].

The main recent development in the determination of parton distributions has been the availability of sets of parton distributions with errors [4–6]. Previously, errors on PDFs were crudely estimated by varying some sets of parameters, or comparing different determinations, and generally considered to be negligible in comparison to other sources of theoretical or experimental error. Now, the simultaneous progress in higher-order theoretical calculations and experimental results has made this simple-minded approach obsolete: a *bona fide* estimate of the error on PDFs has become necessary.

This is a difficult problem, not only because of the usual difficulties in accounting properly for theoretical uncertainties (such as higher order perturbative corrections) which are non-gaussian and hard to estimate, and in collecting and propagating uncertainties contained in large experimental covariance matrices, but also because a PDF set is a set of functions, and therefore one is faced with the problem of constructing an error — a probability measure — on a space of functions [7]. This is clearly an ill-posed problem, because one is trying to infer an infinite amount of information from a finite set of data points: it can be made tractable only by introducing some theoretical assumptions.

The standard approach to this problem is based on the choice of a specific functional form: the infinite-dimensional problem of determining a function is projected onto the finite-dimensional space of parameters which determine the given parametrization. Errors on PDFs are then essentially error ellipsoids in the finite-dimensional parameter space, which at least in principle can be determined by standard covariance matrix techniques. Based on this approach, three sets of PDFs with errors have been produced over the last few years [4–6]. The most obvious shortcoming of this sort of approach is that the choice of a parametrization is potentially a source of bias. More subtle problems are related to non-gaussian errors (either theoretical and experimental) and incompatible data, which render a simple maximum likelihood fit impossible or unreliable.

These difficulties are apparent in the available sets of PDFs with errors, especially when they are compared with each other. Indeed, the direct determination of error bands as one-sigma contours seems to be possible only if a limited set of data is fitted. In particular, straightforward determination of errors as one-sigma contours has succeeded

in a global PDF fit to deep-inelastic data only [4], but as soon as new data (specifically from the Drell-Yan process) are added to this fit, their errors must be suitably rescaled [2]. Global fits to all available data are instead typically constructed by determining a priori a suitable “tolerance criterion”, by inspection of the quality of the fit to all available data [5]. For example, one may determine the tolerance criterion by considering the spread of 90% confidence intervals for various experiments, as one moves away from the minimum of the χ^2 along eigenvectors of the Hessian matrix, and taking the envelope of the resulting ranges. In practice, this suggests that a reasonable estimate of the one-sigma contours for PDFs can be obtained by selecting an appropriately large value of $\Delta\chi^2$, such as $\Delta\chi^2 = 100$ [5,6]. Even so, results obtained using different parton sets for relevant physical observables, such as W+Higgs production at the LHC, are sometimes found to disagree within their stated error bands [2,8]. The origin of these difficulties is most likely a combination of several factors: inconsistencies between different pieces of experimental data, underestimation of some experimental errors, bias due to specific choices of parton parametrization, and non-gaussian errors, especially in the space of parameters of individual parton parametrizations.

These difficulties have stimulated various proposals for new approaches to the determination of parton distributions, in particular with the aim of minimizing the bias related to the choice of a specific functional form, and trying instead to reconstruct the probability density in the full functional space of parton distributions. An interesting suggestion [7] is to use Bayesian inference based on the data in order to update the existing prior knowledge of PDFs, as summarized e.g. by a Monte Carlo sample based on an existing parton set. Although promising preliminary results have been obtained, no parton set based on this approach has been published yet.

An alternative suggestion has been presented in Ref. [9]. The basic idea is to combine a Monte Carlo sampling of the probability measure on the space of functions that one is trying to determine (as in the approach of Ref. [7]) with the use of neural networks as universal unbiased interpolating functions. In a Monte Carlo approach, the function with error — the up quark distribution, say — is given as a Monte Carlo sample of replicas of the function, so that any statistical property of the underlying distribution can be derived from the given sample. For example, the average value of the function at some point is simply found as the average over the replicas, its error as the variance and so on. In a neural network approach, each function in the sample in turn is given by a neural network which, when fed the values of the kinematic variables as input, returns the value of the function itself as output. The underlying idea is that neural networks can be used as universal unbiased interpolators: starting from a Monte Carlo representation of the probability density at the points where it is known because there are data, they can be used to produce a representation of the probability density everywhere in the data region, and even to study the extrapolation outside it, and its breakdown.

In Refs. [9,10] this strategy was tried off on a somewhat simpler problem, namely, the construction of a parametrization of existing data on the deep-inelastic structure function $F_2(x, Q^2)$ of the proton and neutron. In such case, one is only testing that the method can be used to construct a faithful representation of the probability density in a space of functions, based on the measurement of the function at a finite discrete number of points.

The method was proven to be fast and robust, to be amenable to detailed statistical studies, and to be in many respects superior to conventional parametrizations of structure functions based on a fixed functional form. However, the determination of a parton set involves the significant complication of having to go from one or more physical observables to a set of parton distributions. In order to achieve the result, one must tackle various problems, such as deconvolution of hard coefficient functions and resolution of evolution equations, and several further subtleties such as the treatment of higher twist corrections and heavy quark thresholds. All these issues are of course rather well understood in the context of parton determinations, but their implementation in a neural Monte Carlo framework is nontrivial, both in principle and in practice.

In this paper, we present a first determination of the nonsinglet quark distribution from deep-inelastic data based on neural networks. This result is the first step towards the determination of a full parton distribution set. Furthermore, almost all of the technical complications required for the construction of a neural parton arise and have to be tackled already in the nonsinglet case. Hence, this paper is a comprehensive introduction to neural parton fitting.

The paper is organized as follows. In Sect. 2 we review the approach of Refs. [9,10] to the construction of a neural network parametrization of a function based on its experimental sampling at a finite set of points, we describe its application to the construction of a parton set, thereby outlining the general strategy of the method, and we summarize the main features of our approach and thus of our results. In Sect. 3 we construct a Monte Carlo representation of the information contained in the data, and test for its statistical accuracy. In Sect. 4 we discuss a novel strategy to determine the evolution of parton distributions and to combine them with perturbative coefficients in order to determine the structure functions, which combines the speed and reliability of Mellin N -space evolution methods with the flexibility of x -space evolution methods with respect to the choice of parton parametrization. In Sect. 5 we turn to the neural networks which are used to parametrize parton distributions, we discuss their features, the training procedure whereby they are fitted to the data, and the way this training is stopped before the noise in the data starts affecting the results of the fit. In Sect. 6 we present our reference next-to-leading order determination of the nonsinglet quark distribution, we test for the accuracy of our estimate of central value and statistical uncertainty, and we verify the stability of our result upon the minimization strategy. We discuss in particular how we arrive at an accurate determination of the statistical properties (specifically the uncertainty) of this determination. Potential sources of theoretical uncertainty, in particular higher twists and higher orders, are discussed in Sect. 7, while in Sect. 8 our reference result is compared to results obtained at different perturbative orders (leading and next-to-next-to-leading) and to alternative existing determinations.

2. General strategy

The method developed in Ref. [9] provides a general framework for the parametrization of physical observables by means of neural networks. This method has since then been

successfully applied to a diverse variety of physical problems: structure functions [10], spectral functions for τ decays [11], energy spectra of B decays [12], and cosmic ray neutrino fluxes [13], thereby proving its flexibility and robustness. Its application to parton distributions is based on the same underlying concepts, but it is significantly more intricate for a variety of reasons to be discussed shortly, the most obvious being that parton distributions are not directly physical observable quantities.

The general underlying strategy of this approach is summarized in the flow-chart of Fig. 1, and, as in Ref. [9], it involves two distinct stages in order to go from the data to the parton parametrization. In the first stage, a Monte Carlo sample of replicas of the data (“artificial data”) is generated. These can be viewed as a sampling of the probability measure on the space of physical observables (structure functions, cross sections, etc.) at the discrete points where data exist. In the second stage one uses neural networks to interpolate between points where data exist. When constructing a parton set, this second stage in turn consists of two sub-steps: the determination of physical observables from parton distributions (“evolution”), and the comparison of the physical observables thus computed to the data in order to tune the best-fit form of input neural parton distribution (“training of the neural network”). Combining these two steps, the space of physical observables is mapped onto the space of parton distributions, so the experimental information on the former can be interpolated by neural networks in the latter. Let us now describe each stage in turn, both in general and in the specific case discussed in this paper.

The starting experimental data in general consist of the determination of several physical observables in distinct experiments, each of which provides the measurement of the relevant quantity at a discrete set of values of the kinematical variables. In general, there will be a nontrivial set of correlations between determination of different quantities: e.g., different observables determined at different points may be correlated both between observables and between points. In the current nonsinglet fit, we will consider a single observable, namely the nonsinglet nucleon structure function defined as

$$F_2^{\text{NS}}(x, Q^2) \equiv F_2^p(x, Q^2) - F_2^d(x, Q^2) , \quad (2.1)$$

in terms of the proton and deuteron structure functions.

The purpose of the artificial data generation is to produce a Monte Carlo set of ‘pseudo-data’, i.e. N_{rep} replicas of the original set of N_{dat} data points:

$$F_i^{\text{(art)}(k)}; \quad k = 1, \dots, N_{\text{rep}}, \quad i = 1, \dots, N_{\text{dat}}, \quad (2.2)$$

where F_i denotes one individual measurement: in our specific case, a measurement of $F_2^{\text{NS}}(x_i, Q_i^2)$. The N_{rep} sets of N_{dat} points are distributed according to an N_{dat} -dimensional multi-gaussian distribution about the original points, with expectation values equal to the central experimental values, and error and covariance equal to the corresponding experimental quantities. Because the distribution of the experimental data coincides (for a flat prior) with the probability distribution of the value of the structure function at the points where it has been measured [14, 15], this Monte Carlo set gives a sampling of the probability measure at those points. We can then generate arbitrarily many sets of pseudo-data,

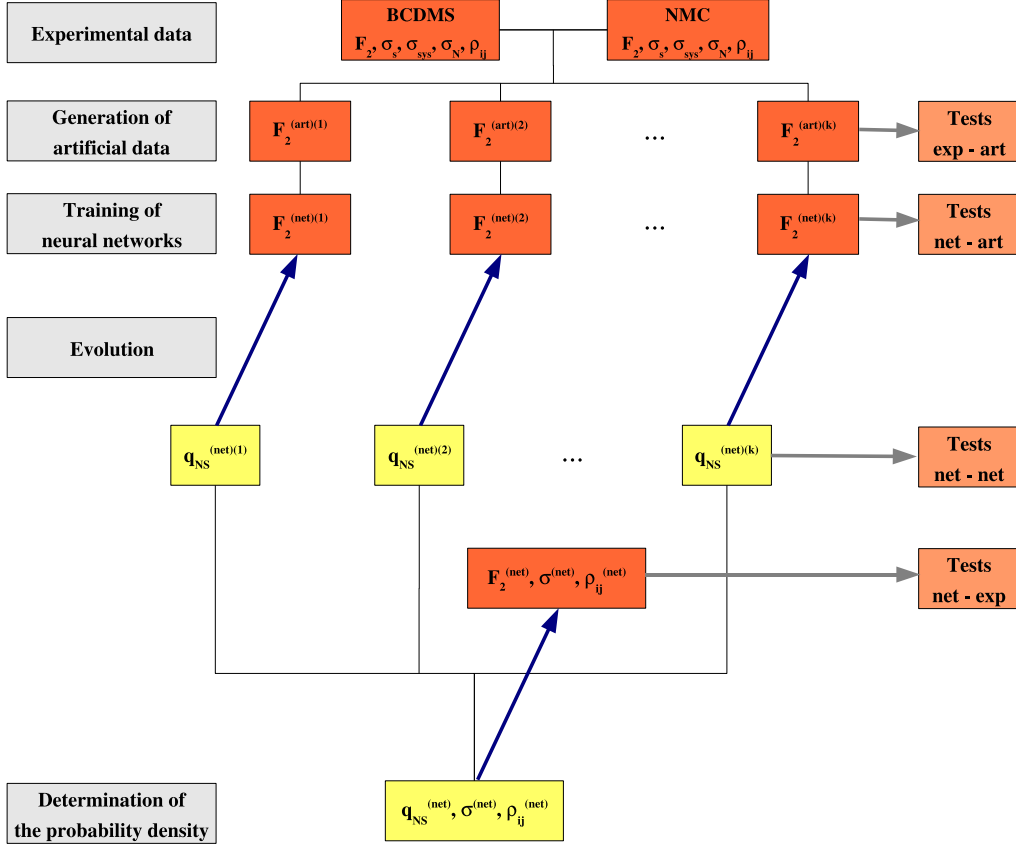


Figure 1: Flow-chart for the construction of a neural parton set.

and choose the number of sets N_{rep} in such a way that the properties of the Monte Carlo sample reproduce those of the original data set to arbitrary accuracy. For example, we can check that averages, variance and covariance of the pseudo-data reproduce central values and covariance matrix elements of the original data. This step is denoted by 'Tests exp-art' in Fig. 1. Note that experimental errors (including correlated systematics) must be treated as gaussian since this is what experimental collaborations provide: the relevant issue is whether an underlying non-gaussian distribution of best-fit parton distributions can be obtained as a result of the fitting procedure. Whereas this might be problematic if errors are determined as one-sigma contours for a given parametrization, the Monte Carlo method gives considerably more flexibility, as we now show.

The second step consists of training N_{rep} sets of neural networks. Each set contains all the parton distributions which are being determined, as a function of x at a given reference scale Q_0^2 , and is based on all the data in one single replica of the original data set. Thus, each set will contain N_{pdf} parton distributions, determined from N_{dat} data. The number of independent parton distributions is $1 \leq N_{pdf} \leq 13$, and it will depend on the specific data set which is used in the parton determination. At most, there can be six quark and antiquark distributions and one gluon distribution, but in practice not all of these may be

accessible with a given set of data. Specifically, in the nonsinglet fit discussed in this paper we will consider the simplest possible case, in which each set contains only one parton distribution, namely the C-even nonsinglet quark distribution

$$q_{\text{NS}}(x, Q_0^2) \equiv [u(x, Q_0^2) + \bar{u}(x, Q_0^2) - (d(x, Q_0^2) + \bar{d}(x, Q_0^2))]. \quad (2.3)$$

As in any parton fit, in order to compare the parton distributions with the data, the PDFs must be evolved from the reference scale Q_0^2 to the scale at which data are given, and combined with hard partonic cross sections in order to obtain physical observables. Unlike in most parton fits (such as e.g. [4–6]), however, each parton distribution is parametrized by a neural network, rather than by a fixed functional form. Then, the N_{dat} data in each replica are used to determine the N_{pdf} neural networks of the corresponding set. At the end of the procedure, we end up with N_{rep} sets of N_{pdf} parton distributions, with each PDF given by a neural network. The N_{rep} replicas of each parton distribution provide the corresponding probability density: for example, the mean value of the parton distribution at the starting scale for a given value of x is found by averaging over the replicas, and the uncertainty on this value is the variance of the values given by the replicas.

The fit of the parton distributions to each replica of the data is performed by maximum likelihood, by minimizing an error function, which coincides with the χ^2 of the experimental points when compared to their theoretical determination obtained using the given set of parton distributions. The χ^2 is computed by fully including the covariance matrix of the correlated experimental uncertainties (with a suitable treatment of normalization errors [14]), as we shall discuss in more detail in Sect. 5.1 below. Unlike in conventional determinations of parton distributions with errors [4–6], however, the covariance matrices of the best-fit parameters are irrelevant and need not be computed. The uncertainty on the final result is found from the variance of the Monte Carlo sample. This eliminates the problem of choosing the value of $\Delta\chi^2$ which corresponds to a one-sigma contour in the space of parameters, a problem which becomes nontrivial in the presence of incompatible data, underestimated errors, or unknown systematics, and which therefore plagues current global fits [5, 6].

Rather, one only has to make sure that each neural network provides a consistent fit to its corresponding replica. If the underlying data are incompatible or have underestimated errors, the best fit might be worse than one would expect with properly estimated gaussian errors — for instance in the presence of underestimated errors it will have typically a value of χ^2 per degree of freedom larger than one. If, on the contrary, experimental errors have been overestimated, the final χ^2 can turn out to be smaller than one per degree of freedom. Neural networks are ideally suited for providing a fit in this situation, based on the reasonable assumption of smoothness: for example, incompatible data or data with underestimated errors will naturally be fitted less accurately by the neural network [9]. Also, this allows for non-gaussian behavior of experimental uncertainties. Indeed, whereas a neural network with sufficiently large architecture can fit any compatible set of data, the choice of a suitable stopping criterion ensures that only the information contained in the data is fitted, but not the noise. This can be done by verifying that the quality of the fit

of data which have been used in the fitting procedure is the same as the quality of the fit to data which *have not* been used for fitting.

It is then possible to verify that each neural network has the correct statistical features. This step is denoted by 'Tests net-art' in Fig. 1. Finally, the self-consistency of the Monte Carlo sample can be tested in order to ascertain that it leads to consistent estimates of the uncertainty on the final parton distributions, for example by verifying that the value of the parton distribution extracted from different replicas indeed behaves as a random variable with the stated variance. This step is denoted by 'Tests net-net' in Fig. 1. This set of tests allow us to make sure that the Monte Carlo sample of neural networks provides a faithful and consistent representation of the information contained in the data on the probability measure in the space of parton distributions, and in particular that the value of parton distributions and their correlated uncertainties are correctly estimated.

Because of the use of neural networks for the parametrization of the parton distributions, two technical aspects of our fitting procedure differ from what is done in standard parton fits. The first is the method which is used in order to evolve parton distributions by solving the QCD evolution equations. Indeed, we solve evolution equations in Mellin moment space N [16], since this allows for a fast and accurate numerical solution. However, this method is usually applied to parton parametrizations such that the Mellin transform of the initial parton distribution can be computed analytically: a numerical approach is not viable because in the region where the inverse Mellin transform of the solution exists, the Mellin transform of the initial condition does not converge. But a neural network is a complicated nonlinear function whose Mellin transform cannot be determined analytically in closed form. The alternative of parametrizing directly the initial parton distributions in N moment space is unpalatable, because one would need a parametrization in the complex plane of the N variable, whereas neural networks are most conveniently defined on a compact space, such as $0 \leq x \leq 1$.

Therefore, we are led to introduce a novel technique, which consists of determining the x -space evolution kernel (Green function) from the N space solution, in such a way that any parton distribution can be evolved by convolution with this universal kernel. This method combines the speed and reliability of the N -space solution method, with the advantages of parametrizing the parton distribution in x space. Furthermore, because the kernel is universal (i.e. independent of the specific boundary condition which is evolved) it can be computed beforehand to arbitrarily high accuracy and stored; the result can then be used during the fitting procedure, thus leading to an extremely efficient evolution. This evolution method will be discussed in detail in Sect. 4.

The other technical peculiarity which is motivated by the use of neural networks is the choice of the minimization algorithm. Because of the nonlinear dependence of the neural network on its parameters, and the nonlocal dependence of the measured quantities on the neural network (cross sections are expressed as convolution of the initial distribution with evolution kernels and coefficient functions), a genetic algorithm turns out to be the most efficient minimization method. The use of a genetic algorithm is particularly convenient when seeking a minimum in a very wide space with potentially many local minima, because the method handles a population of solutions rather than traversing a path in the space of

solutions. The minimization method will be discussed in detail in Sect. 5.

At the end of the minimization we end up with a Monte Carlo sample of neural networks which provides our best representation of the measure in the space of parton distribution. Generally, we expect [9, 10] the uncertainty on our final result to be somewhat smaller than the uncertainty on the input data, because the information contained in several data points is combined. The properties of this measure can be tested against the input data by using it to compute means, variance and covariances which can be compared to the input experimental ones which have been used in the parton determination. Also, the stability of the result can be tested by excluding some data points from the fit and checking that the best-fit result predicts them correctly. These tests are denoted by 'Tests net-exp' in Fig. 1, and they conclude our determination.

3. Experimental data and Monte Carlo generation

3.1 Experimental data and kinematical cuts

The nonsinglet parton distribution $q_{\text{NS}}(x, Q_0^2)$ can be extracted from experimental data on proton and deuteron structure functions $F_2^p(x, Q^2)$ and $F_2^d(x, Q^2)$. A detailed discussion of all the available data was given in Refs. [9, 10]. In short, early SLAC data have extremely large uncertainties, while data from JLAB and the E665 collaboration [17] are mostly at low Q^2 and are almost entirely excluded by our kinematic cuts. This leaves us with the data published by the NMC [18] and BCDMS [19, 20] collaborations. We refer to [9] for a discussion of the kinematical and statistical features of these data, in particular for the details of their correlated systematics.

Once the systematics are known, the experimental covariance matrix for each experiment can be easily computed

$$\text{cov}_{ij} = \left(\sum_{p=1}^{N_{\text{sys}}} \sigma_{i,p} \sigma_{j,p} + F_i F_j \sigma_N^2 \right) + \delta_{ij} \sigma_{i,s}^2, \quad (3.1)$$

where F_i, F_j are central values, $\sigma_{i,p}$ are the N_{sys} correlated systematic errors, σ_N is the total normalization uncertainty, and $\sigma_{i,s}$ is the statistical uncertainty. The correlation matrix is given by

$$\rho_{ij} = \frac{\text{cov}_{ij}}{\sigma_{i,\text{tot}} \sigma_{j,\text{tot}}}, \quad (3.2)$$

where the total error $\sigma_{i,\text{tot}}$ for the i -th point is given by

$$\sigma_{i,\text{tot}} = \sqrt{\sigma_{i,s}^2 + \sigma_{i,c}^2 + F_i^2 \sigma_N^2} \quad (3.3)$$

and the total correlated uncertainty $\sigma_{i,c}$ is the sum of all correlated systematics

$$\sigma_{i,c}^2 = \sum_{p=1}^{N_{\text{sys}}} \sigma_{i,p}^2. \quad (3.4)$$

In order to keep higher twist corrections under control, only data with $Q^2 > 3 \text{ GeV}^2$ are retained. No separate cut in the invariant mass of the final state W^2 turns out to be

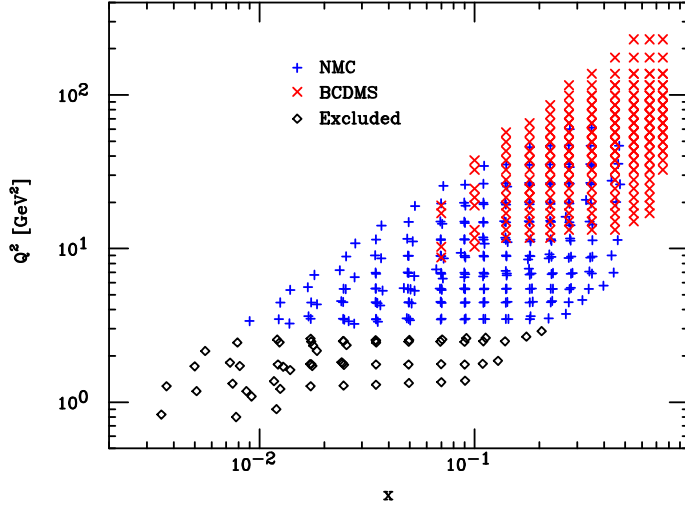


Figure 2: Kinematical coverage of the available experimental data on the nonsinglet structure function in the (x, Q^2) plane.

Experiment	x range	Q^2 range	N_{dat}	$\langle \sigma_s \rangle$	$\langle \sigma_c \rangle$	$\langle \frac{\sigma_c}{\sigma_s} \rangle$	$\langle \sigma_N \rangle$	$\langle \sigma_{\text{tot}} \rangle$	$\langle \rho \rangle$	$\langle \text{cov} \rangle$
NMC	$9.0 \cdot 10^{-3} - 4.7 \cdot 10^{-1}$	$3.2 - 61 \text{ GeV}^2$	229	103.2	85.6	0.75	72.5	157.5	0.038	0.199
BCDMS	$7.0 \cdot 10^{-2} - 7.5 \cdot 10^{-1}$	$8.7 - 230 \text{ GeV}^2$	254	22.3	12.2	0.50	22.3	37.3	0.163	0.085

Table 1: Experimental data included in this analysis. All values of σ and cov are given as percentages. Averages over all data points are noted with brackets $\langle \rangle$.

necessary. A discussion on the motivation for this choice of cuts and the possible effect of their variation will be given in Sect. 6.

A scatter plot of the NMC and BCDMS data in the (x, Q^2) plane is displayed in Fig. 2, with points excluded due to the Q^2 cut marked with diamonds. Note that all these excluded data points belong to the NMC experiment. In Table 1, the statistical features of the data (after the cut) are summarized.

3.2 Monte Carlo generation

The first step in our approach, as discussed in Sect. 2, is to generate a Monte Carlo sample of replicas of the experimental data. Each Monte Carlo replica of the original experimental data is generated as a multi-gaussian distribution. More precisely, for each data point $F_i^{(\text{exp})} \equiv F_2^{\text{NS}}(x_i, Q_i^2)$ we generate $k = 1, \dots, N_{\text{rep}}$ artificial points $F_i^{(\text{art})^{(k)}}$ as follows

$$F_i^{(\text{art})^{(k)}} = \left(1 + r_N^{(k)} \sigma_N\right) \left(F_i^{(\text{exp})} + \sum_{p=1}^{N_{\text{sys}}} r_p^{(k)} \sigma_{i,p} + r_i^{(k)} \sigma_{i,s}\right), \quad k = 1, \dots, N_{\text{rep}}, \quad (3.5)$$

using independent univariate gaussian random numbers $r^{(k)}$ for each independent error source.

The value of N_{rep} is determined in such a way that the Monte Carlo set of replicas models faithfully the probability distribution of the data in the original set. A quantitative comparison can be performed by defining suitable statistical estimators (see Appendix

$$F_2^{\text{NS}}(x, Q^2)$$

N_{rep}	10	100	1000
$\langle PE [\langle F^{(\text{art})} \rangle_{\text{rep}}] \rangle$	20%	6.4%	1.3%
$r [F^{(\text{art})}]$	0.97	0.99	0.99
$\langle V [\sigma^{(\text{art})}] \rangle_{\text{dat}}$	$6.1 \cdot 10^{-5}$	$1.9 \cdot 10^{-5}$	$6.7 \cdot 10^{-6}$
$\langle PE [\sigma^{(\text{art})}] \rangle_{\text{dat}}$	33%	11%	3%
$\langle \sigma^{(\text{art})} \rangle_{\text{dat}}$	0.011	0.011	0.011
$r [\sigma^{(\text{art})}]$	0.94	0.99	0.99
$\langle V [\rho^{(\text{art})}] \rangle_{\text{dat}}$	0.10	$9.4 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
$\langle \rho^{(\text{art})} \rangle_{\text{dat}}$	0.182	0.097	0.100
$r [\rho^{(\text{art})}]$	0.47	0.79	0.97
$\langle V [\text{cov}^{(\text{art})}] \rangle_{\text{dat}}$	$5.5 \cdot 10^{-9}$	$1.7 \cdot 10^{-10}$	$5.7 \cdot 10^{-11}$
$\langle \text{cov}^{(\text{art})} \rangle_{\text{dat}}$	$1.3 \cdot 10^{-5}$	$7.6 \cdot 10^{-6}$	$8.1 \cdot 10^{-6}$
$r [\text{cov}^{(\text{art})}]$	0.41	0.81	0.98

Table 2: Comparison between experimental and Monte Carlo data.

The experimental data have $\langle \sigma^{(\text{exp})} \rangle_{\text{dat}} = 0.011$, $\langle \rho^{(\text{exp})} \rangle_{\text{dat}} = 0.107$ and $\langle \text{cov}^{(\text{exp})} \rangle_{\text{dat}} = 8.6 \cdot 10^{-6}$.

B). Results are presented in Table 2. The results show that a sample of 1000 replicas is sufficient to ensure average scatter correlations of 99% and accuracies of a few percent on structure functions, errors and correlations.

Normalization errors are not included in the covariance matrix on the same footing as other sources of systematics since, as by now well known [14], this would bias the fit. Rather, normalization errors are included by rescaling all errors (statistical $\sigma_{i,s}$ and each systematic $\sigma_{i,p}$) independently for each replica by a factor

$$\begin{aligned} \overline{\sigma}_{i,s}^{(k)} &= (1 + r_N^{(k)} \sigma_N) \sigma_{i,s} , \\ \overline{\sigma}_{i,p}^{(k)} &= (1 + r_N^{(k)} \sigma_N) \sigma_{i,p} \quad p = 1, \dots, N_{\text{sys}} , \end{aligned} \quad (3.6)$$

where $r_N^{(k)}$ is the random variable Eq. (3.5) associated to the normalization uncertainty. The covariance matrix is then given by

$$\overline{\text{cov}}^{(k)}_{ij} = \left(\sum_{p=1}^{N_{\text{sys}}} \overline{\sigma}_{i,p}^{(k)} \overline{\sigma}_{j,p}^{(k)} \right) + \delta_{ij} \overline{\sigma}_{i,s}^{(k)2} , \quad (3.7)$$

in terms of the rescaled uncertainties.

4. From parton distributions to physical observables

The nonsinglet structure function $F_2^{\text{NS}}(x, Q^2)$ Eq. (2.1) is determined from the nonsinglet quark distribution $q_{\text{NS}}(x, Q_0^2)$ Eq. (2.3) at a reference scale Q_0^2 by first evolving the parton distribution to the scale Q^2 by means of QCD evolution equations, and then convoluting

it with the appropriate hard coefficient function $C_{\text{NS}}(x, \alpha_s(Q^2))$:

$$F_2^{\text{NS}}(x, Q^2) = \frac{1}{6} x \int_x^1 \frac{dy}{y} C_{\text{NS}}(y, \alpha_s(Q^2)) q_{\text{NS}}\left(\frac{x}{y}, Q^2\right). \quad (4.1)$$

Both coefficient functions and the anomalous dimensions which drive perturbative evolution are now known up to next-to-next-to leading order (and in fact the coefficient function up to NNNLO [21]). Therefore, we will provide determinations of the parton distributions at leading (LO), next-to-leading (NLO) and next-to-next-to leading order (NNLO). In this section we will discuss in detail the methods that we use in order to actually determine the structure function from the input parton distribution, specifically the resolution of evolution equations. Only the nonsinglet case will be discussed explicitly, while the (straightforward) generalization to the singlet case will be discussed in a forthcoming publication.

4.1 Leading-twist evolution and coefficient functions

As well known, QCD evolution equations are most easily written for Mellin moments of the parton distributions, defined as

$$q_{\text{NS}}(N, Q^2) \equiv \int_0^1 dx x^{N-1} q_{\text{NS}}(x, Q^2), \quad (4.2)$$

where by slight abuse of notation we denote the function and its transform with the same symbol. They take the form

$$\frac{d}{d \ln Q^2} q_{\text{NS}}(N, Q^2) = \frac{\alpha_s(Q^2)}{4\pi} \gamma_{\text{NS}}(N, \alpha_s(Q^2)) q_{\text{NS}}(N, Q^2), \quad (4.3)$$

where the anomalous dimension can be expanded in powers of $\alpha_s(Q^2)$ as

$$\gamma_{\text{NS}}(N, \alpha_s(Q^2)) = \gamma_{\text{NS}}^{(0)}(N) + \frac{\alpha_s(Q^2)}{4\pi} \gamma_{\text{NS}}^{(1)}(N) + \left(\frac{\alpha_s(Q^2)}{4\pi}\right)^2 \gamma_{\text{NS}}^{(2)}(N) + \dots \quad (4.4)$$

The NNLO contribution has been computed recently in Ref. [22]. Defining analogously the Mellin transforms of coefficient function and structure function, the latter is given by

$$F_2^{\text{NS}}(N-1, Q^2) = \frac{1}{6} C_{\text{NS}}(N, \alpha_s(Q^2)) q_{\text{NS}}(N, Q^2), \quad (4.5)$$

with the perturbative expansion of the coefficient function

$$C_{\text{NS}}(N, \alpha_s(Q^2)) = 1 + \frac{\alpha_s(Q^2)}{4\pi} C_{\text{NS}}^{(1)}(N) + \left(\frac{\alpha_s(Q^2)}{4\pi}\right)^2 C_{\text{NS}}^{(2)}(N) + \dots \quad (4.6)$$

It appears therefore especially convenient [16, 23] to determine the structure function in Mellin space and then invert the Mellin transform, since the N space expression of the structure function can be determined in closed form, and the problem is reduced to the computation of the single Mellin inversion integral. In x -space instead one has to solve an integro-differential equation. Even though efficient numerical methods have been

developed for the resolution of the integro-differential form of the evolution equation (see e.g. [24, 25]), the only genuine advantage of an x -space approach is the possibility to work directly with an x -space parametrization of parton distributions. As discussed in Sect. 2, this option is especially convenient when using neural networks, since these are most easily defined as function on a compact space such as $0 \leq x \leq 1$.

However, we observe that the solution to the evolution equation (4.3) has the form

$$q_{\text{NS}}(N, Q^2) = \Gamma(N, \alpha_s(Q^2), \alpha_s(Q_0^2)) q_{\text{NS}}(N, Q_0^2), \quad (4.7)$$

where the evolution kernel $\Gamma(N, \alpha_s(Q^2), \alpha_s(Q_0^2))$ is entirely determined by the anomalous dimension $\gamma_{\text{NS}}(N, \alpha_s(Q^2))$, and it does not depend on the boundary condition $q_{\text{NS}}(N, Q_0^2)$. Hence, if we take an inverse Mellin transform of Eq. (4.7) we get

$$q_{\text{NS}}(x, Q^2) = \int_x^1 \frac{dy}{y} \Gamma(y, \alpha_s(Q^2), \alpha_s(Q_0^2)) q_{\text{NS}}\left(\frac{x}{y}, Q_0^2\right). \quad (4.8)$$

We can then determine the kernel $\Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2))$ explicitly by Mellin inversion, and use it to evolve any x -space parton distribution using Eq. (4.8). The only complication is related to the large- x behavior of the kernel $\Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2))$.

Furthermore, we can express the structure function in terms of the initial quark distribution as

$$F_2^{\text{NS}}(N-1, Q^2) = \frac{1}{6} \tilde{\Gamma}(N, \alpha_s(Q^2), \alpha_s(Q_0^2)) q_{\text{NS}}(N, Q_0^2), \quad (4.9)$$

where

$$\tilde{\Gamma}(N, \alpha_s(Q^2), \alpha_s(Q_0^2)) \equiv C_{\text{NS}}(N, \alpha_s(Q^2)) \Gamma(N, \alpha_s(Q^2), \alpha_s(Q_0^2)), \quad (4.10)$$

i.e. absorbing the coefficient function in a redefined evolution kernel. Again, we can determine the Mellin inverse of this redefined kernel, and use it to obtain the measured structure function from the initial x -space parton distribution as

$$F_2^{\text{NS}}(x, Q^2) = \frac{1}{6} x \int_x^1 \frac{dy}{y} \tilde{\Gamma}(y, \alpha_s(Q^2), \alpha_s(Q_0^2)) q_{\text{NS}}\left(\frac{x}{y}, Q_0^2\right). \quad (4.11)$$

The evolution kernels can be computed, interpolated to arbitrary accuracy and stored, and then used to evolve the initial PDF or determine the structure function from it through the computation of a single real convolution integral, eqs. (4.8) and (4.11).

4.2 Solution of the evolution equation in N space

We determine the evolution kernel up to NNLO in terms of the running coupling, for which we take the expanded solution of the renormalization group equation, i.e., up to NNLO

$$\begin{aligned} \alpha_s(Q^2) = & \alpha_s(Q^2)_{LO} \left[1 + \alpha_s(Q^2)_{LO} [\alpha_s(Q^2)_{LO} - \alpha_s(M_Z^2)] (b_2 - b_1^2) \right. \\ & \left. + \alpha_s(Q^2)_{NLO} b_1 \ln \frac{\alpha_s(Q^2)_{NLO}}{\alpha_s(M_Z^2)} \right], \end{aligned} \quad (4.12)$$

with

$$\alpha_s(Q^2)_{NLO} = \alpha_s(Q^2)_{LO} \left[1 - b_1 \alpha_s(Q^2)_{LO} \ln \left(1 + \beta_0 \alpha_s(M_Z^2) \ln \frac{Q^2}{M_Z^2} \right) \right], \quad (4.13)$$

$$\alpha_s(Q^2)_{LO} = \frac{\alpha_s(M_Z^2)}{1 + \beta_0 \alpha_s(M_Z^2) \ln \frac{Q^2}{M_Z^2}}, \quad (4.14)$$

and the beta function coefficients given by

$$Q^2 \frac{d\alpha_s(Q^2)}{dQ^2} = - \sum_{k=0}^2 \beta_k \alpha_s(Q^2)^{k+2}, \quad a_s(Q^2) = \frac{\alpha_s(Q^2)}{4\pi}, \quad (4.15)$$

where

$$\beta_0 = 11 - \frac{2}{3} N_f, \quad (4.16)$$

$$\beta_1 = 102 - \frac{38}{3} N_f, \quad (4.17)$$

$$\beta_2 = \frac{2857}{2} - \frac{5033}{18} N_f + \frac{325}{54} N_f^2, \quad (4.18)$$

and $b_i \equiv \beta_i / \beta_0$. We will use the current PDG [26] average value of $\alpha_s(Q^2)$, namely

$$\alpha_s(M_Z^2) = 0.118 \pm 0.002 \quad (4.19)$$

with $M_Z = 91.187$ GeV.

The evolution factor is then explicitly given by

$$\begin{aligned} \Gamma(N, \alpha_s(Q^2), \alpha_s(Q_0^2)) &= \left(\frac{\alpha_s(Q^2)}{\alpha_s(Q_0^2)} \right)^{-\gamma^{(0)}(N)/\beta_0} \left(1 - (a_s(Q^2) - a_s(Q_0^2)) U_{\text{NS}}^{(1)}(N) \right. \\ &\quad \left. + (a_s^2(Q^2) - a_s^2(Q_0^2)) U_{\text{NS}}^{(2)}(N) - a_s(Q^2) a_s(Q_0^2) \left(U_{\text{NS}}^{(1)}(N) \right)^2 \right), \end{aligned} \quad (4.20)$$

where (adopting the notations and conventions of Ref. [23]) we have introduced nonsinglet evolution coefficients

$$U_{\text{NS}}^{(1)}(N) = -\frac{1}{\beta_0} \left(\gamma^{(1)}(N) - b_1 \gamma^{(0)}(N) \right), \quad (4.21)$$

$$U_{\text{NS}}^{(2)}(N) = -\frac{1}{2} \left[\frac{1}{\beta_0} \gamma^{(2)}(N) - b_1 U_{\text{NS}}^{(1)} - \frac{b_2}{\beta_0} \gamma^{(0)}(N) - \left(U_{\text{NS}}^{(1)}(N) \right)^2 \right]. \quad (4.22)$$

At NNLO, both the strong coupling and the parton distributions are discontinuous when crossing heavy quark thresholds. We set the thresholds $m_c = 1.4$ GeV, $m_b = 4.5$ GeV and $m_t = 175$ GeV with, at NNLO, the discontinuity

$$\alpha_{s,f+1}(m_f^2) = \alpha_{s,f}(m_f^2) + \left(\frac{C_2}{4\pi} \right)^2 \alpha_{s,f}(m_f^2)^3, \quad (4.23)$$

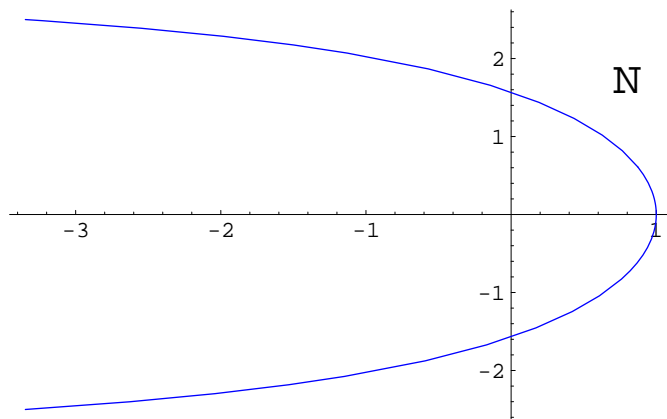


Figure 3: The path in the complex N -space followed by the Talbot integration path, Eq. (4.26) for $r = 1$.

where $C_2 = 14/3$ [27]. We match parton distributions at threshold as

$$q_{\text{NS}}^{(n_f+1)}(N, m_h^2) = q_{\text{NS}}^{(n_f)}(N, m_h^2) . \quad (4.24)$$

Heavy quark mass terms are included in the hard coefficients, and in the nonsinglet sector they vanish up to NLO, while at NNLO they are not known (though they can be approximated from their large x behavior [28]). Therefore, we defer the inclusion of heavy quark mass effects to a forthcoming full fit including the singlet contribution, hence, for all practical purposes, we use a zero-mass variable flavour number scheme in the nonsinglet sector. Note that because of this, the nonsinglet structure function F_2^{NS} has a NNLO discontinuity at the heavy quark thresholds.

4.3 Evolution kernel in x -space

The x -space evolution factor is determined by numerical computation of the Mellin inverse

$$\Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2)) = \int_{c-i\infty}^{c+i\infty} \frac{dN}{2\pi i} x^{-N} \Gamma(N, \alpha_s(Q^2), \alpha_s(Q_0^2)) , \quad (4.25)$$

of the N space evolution factor $\Gamma(N, \alpha_s(Q^2), \alpha_s(Q_0^2))$, Eq. (4.20), as well as of the associate $\tilde{\Gamma}(N, \alpha_s(Q^2), \alpha_s(Q_0^2))$, Eq. (4.10). The numerical computation of the Mellin inversion integral is delicate, because the oscillatory behavior of the integrand at large x is not damped by multiplication by an initial PDF. This problem is mitigated by a suitable choice of integration path. We choose the Talbot path, defined by the condition

$$N(\theta) = r\theta(1/\tan\theta + i) , \quad -\pi \leq \theta \leq \pi , \quad (4.26)$$

with r a constant, shown in Fig. 3.

To further improve the numerical efficiency the Fixed Talbot algorithm can be used [29] where the integral is replaced by the sum

$$\Gamma(x) = \frac{r}{M} \left[\frac{1}{2} \Gamma(N=r) x^{-r} + \sum_{k=1}^{M-1} \text{Re} \left[x^{-N(\theta_k)} \Gamma(N(\theta_k)) (1 + i\sigma(\theta_k)) \right] \right] \quad (4.27)$$

where $\sigma(\theta) \equiv \theta + (\theta/\tan\theta - 1)/\tan\theta$, and $\theta_k = k\pi/M$, and $r = 2M/(5\ln 1/x)$. It can be shown that M is the relative accuracy, i.e. the number of accurate digits. We shall take $M = 16$. The x -space evolution kernel $\Gamma(x)$ determined thus is displayed in Fig. 4, at different perturbative orders and in Fig. 5 for different values of Q^2 .

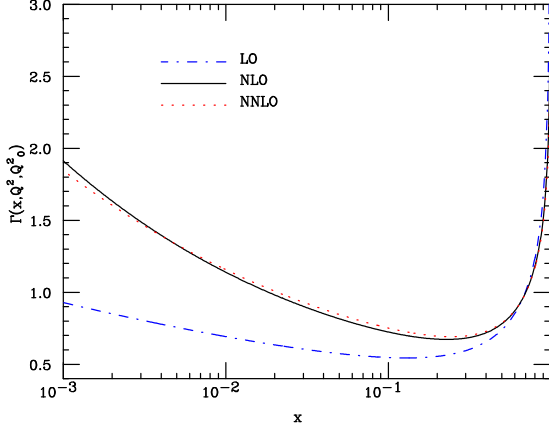


Figure 4: Non-singlet evolution kernel $\Gamma(x)$ computed at different perturbative orders in the kinematical region relevant to nonsinglet evolution. The evolution scales are $Q_0^2 = 2$ GeV² and $Q^2 = 10^4$ GeV².

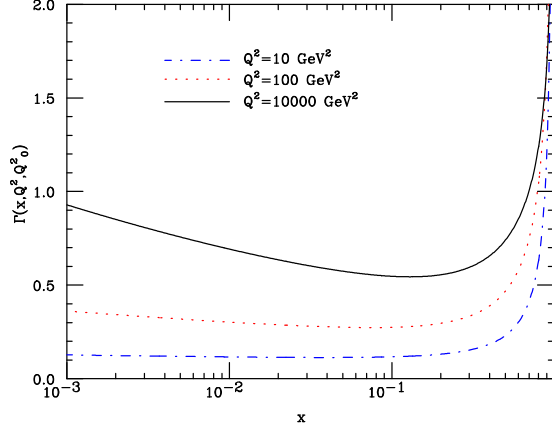


Figure 5: Same as Fig. 4 for the LO dependence of the evolution factor on the evolution length Q^2 . As in the previous case, $Q_0^2 = 2$ GeV².

The determination of the evolution kernel presented so far only holds for $x < 1$. As well known (see Appendix A), the Mellin inverse of the anomalous dimension Eq. (4.4) only exists for $x < 1$, while at $x = 1$ it behaves as a distribution. Because the N -space evolution kernel is obtained by exponentiating the anomalous dimension, its Mellin transform must be defined through its action on a test function. In particular, we must check that integration over the kernel converges as $x \rightarrow 1$. To this purpose, we define a subtracted kernel $\Gamma^{(+)}(x, \alpha_s(Q^2), \alpha_s(Q_0^2))$

$$\Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2)) = \Gamma^{(+)}(x, \alpha_s(Q^2), \alpha_s(Q_0^2)) + G\delta(1-x), \quad (4.28)$$

$$\Gamma^{(+)}(x, \alpha_s(Q^2), \alpha_s(Q_0^2)) \equiv \Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2)) - G\delta(1-x). \quad (4.29)$$

The constant G is defined as

$$G \equiv \int_0^1 dx \Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2)) = \Gamma(N, \alpha_s(Q^2), \alpha_s(Q_0^2)) \Big|_{N=1}, \quad (4.30)$$

where $\Gamma(N, \alpha_s(Q^2), \alpha_s(Q_0^2))$ is given by Eq. (4.20), and it converges at $N = 1$.

The evolution equation, Eq.(4.8), in terms of the subtracted kernel takes the form

$$\begin{aligned} q(x, Q^2) &= Gq(x, Q_0^2) + \int_x^1 \frac{dy}{y} \Gamma^{(+)}(y, \alpha_s(Q^2), \alpha_s(Q_0^2)) q\left(\frac{x}{y}, Q_0^2\right) \\ &= q(x, Q_0^2) \left(G - \int_0^x dy \Gamma(y, \alpha_s(Q^2), \alpha_s(Q_0^2)) \right) \\ &\quad + \int_x^1 \frac{dy}{y} \Gamma(y, \alpha_s(Q^2), \alpha_s(Q_0^2)) \left(q\left(\frac{x}{y}, Q_0^2\right) - yq(x, Q_0^2) \right). \end{aligned} \quad (4.31)$$

It is easy to prove that all integrals in Eq. (4.31) converge and can be computed numerically, exploiting the fact that the behavior of the evolution kernel in the large x limit is known from soft gluon resummation arguments. An explicit proof is given in Appendix A.

Hence we will use Eq. (4.31) to determine the evolution of the quark distribution. The numerical accuracy of this method is tested by comparing it to the benchmark evolution tables first presented in Ref. [30] and recently updated including the full NNLO anomalous dimension in Ref. [2]. In order to allow for a direct comparison to the benchmark tables, we modify some of our final choices in the solution of the evolution equations. Specifically, we fix the number of flavors (to $n_f = 4$) and we keep the NLO contributions to the evolution factor Eq. (4.20) in unexpanded form. With these choices, we determine the evolution of the u valence parton distribution. The results and the accuracy of this benchmark are shown in Table 3, where we use the same parameters as in [2, 30]. More details about the benchmarking of QCD evolution codes can be found in Ref. [2]. We find an accuracy of order $\mathcal{O}(10^{-5})$, comparable to that of current evolution codes. The same result has been obtained for the d valence distribution. Comparable accuracy is expected in C -even evolution, relevant for our paper (not included in the benchmark [30])

The solution of the evolution equation through the determination of an evolution factor is particularly efficient because of the universality of the evolution factor itself, i.e., its independence of the specific boundary condition which is being evolved. Hence, the evolution factor can be precomputed and stored, and then used during the process of parton fitting or when evolving different parton distributions, without having to recompute it each time.

This fact can be exploited in an optimal way during parton fitting and parton evolution. In parton fitting, a given boundary condition must be evolved many times up to the fixed pairs of values of (x, Q^2) at which data are available (in our case, those shown in Fig. 2). Namely, for the i -th data point, one must compute $q(x_i, Q_i^2)$ given by Eq. (4.31). The constant G Eq. (4.30) (which depends only on Q_i^2) can be precomputed and stored for all required values of Q^2 . Furthermore, the numerical determination of the integral over y on the right-hand side of Eq. (4.31) involves the determination of the integrand at a set of values of $y = y_{ki}$, which in turn depend on the given values of x_i, Q_i^2 : $y_{ki} = y_{ki}(x_i, Q_i^2)$. The integrand is fully determined by knowledge of the following values of initial parton distribution and of the evolution kernel:

$$\Gamma_{ik} \equiv \Gamma(y_{ik}, \alpha_s(Q_i^2), \alpha_s(Q_0^2)) \quad (4.32)$$

$$\tilde{q}_{ik} \equiv q\left(\frac{x_i}{y_{ik}}, Q_0^2\right) - y_{ik}q(x_i, Q_0^2). \quad (4.33)$$

The most computing-intensive task is the determination of the values Γ_{ik} Eq. (4.32) evolution kernel. We have precomputed and stored these values for all the necessary values of x_i, Q_i^2 and y_{ik} . In practice, we use gaussian integration with $N_{\text{pt}} = 16(2^{N_{\text{it}}+1} - 1)$ points and determine the values of y accordingly. In Table 4 we show the percentage deviation ϵ of the left-hand side of Eq. (4.31), when compared to what is obtained from the use of a library routine with nominal percentage accuracy of 10^{-5} . It is apparent that $N_{\text{it}} = 4$, i.e. integration with about 500 points is sufficient to reach this accuracy.

This method is of course only convenient when evolving up to a fixed set of points in the (x, Q^2) plane. When evolving up a parton distribution, e.g. in order to input it to some other computation, it is instead convenient to precompute a full interpolation of the kernel $\Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2))$. Evolution is then reduced to the determination of the convolution Eq. (4.31) of this kernel with the parton distribution. This is somewhat less efficient than the previous method because it requires the computation of this convolution over the interpolation each time a point is evolved, but it already avoids the main bottleneck, namely, the Mellin inversion of the kernel. We have interpolated the kernel $\Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2))$ and its integral up to x which appears in the first term of Eq. (4.31), using Chebyshev polynomials.

In order to speed up the interpolation, the leading large x behavior determined at each perturbative order (as described in Appendix A) is divided out before interpolating, which considerably smoothes the large x growth of $\Gamma(x, \alpha_s(Q^2), \alpha_s(Q_0^2))$. We have found that the use of 200 polynomials for the interpolation considered above produces results with a precision of $\mathcal{O}(10^{-5})$, for all the (x, Q^2) range covered by experimental data. This accuracy is enough for present purposes and it could be improved by increasing the number of polynomials. In Table 5 we compare results obtained with the interpolated evolution factors to the exact evolution for different values of x , with the input parton distribution set equal to our best-fit nonsinglet neural parton distribution, to be described in Sect. 6.

We shall choose $Q_0^2 = 2 \text{ GeV}^2$ as a starting scale for perturbative evolution. We have explicitly checked the independence of the results with respect this choice.

4.4 Target mass corrections and higher twists

Even though we discard data at very low Q^2 data in order to minimize the impact of higher twist corrections, target-mass corrections [31], which are the dominant higher twist corrections and are known in closed form should be included in order to increase the accuracy in the low- Q^2 region. The twist-four (i.e. next-to-leading twist) structure function with the inclusion of target-mass corrections is given by

$$F_2^{\text{NLT}}(x, Q^2) = \frac{x^2}{\tau^{3/2}} \frac{F_2^{\text{LT}}(\xi_{\text{TMC}}, Q^2)}{\xi_{\text{TMC}}^2} + 6 \frac{M^2 x^3}{Q^2 \tau^2} I_2(\xi_{\text{TMC}}, Q^2), \quad (4.34)$$

where

$$I_2(\xi_{\text{TMC}}, Q^2) = \int_{\xi_{\text{TMC}}}^1 dz \frac{F_2^{\text{LT}}(z, Q^2)}{z^2}, \quad (4.35)$$

$$\xi_{\text{TMC}} = \frac{2x}{1 + \sqrt{\tau}}, \quad \tau = 1 + \frac{4M^2 x^2}{Q^2}, \quad (4.36)$$

where $F_2^{\text{LT}}(x, Q^2)$ is the leading-twist expression Eq. (4.1).

Fitting directly Eq. (4.34) to the data is impractical, because the nonsinglet quark distribution to be determined appears in the I_2 integral. Rather, it is more convenient to view the contribution proportional to I_2 in Eq. (4.34) as a correction to be applied to the data. Namely, we re-express F_2^{LT} in the I_2 integral in terms of the full F_2 , i.e., to the

next-to-leading twist level we simply replace F_2^{LT} in I_2 with F_2 . Therefore, in practice, the function

$$\Phi_2(x, Q^2) \equiv \frac{x^2}{\tau^{3/2}} \frac{F_2^{\text{LT}}(\xi_{\text{TMC}}, Q^2)}{\xi_{\text{TMC}}^2} \quad (4.37)$$

is fitted to the corrected data:

$$F_2^{\text{dat-TMC}}(x, Q^2) \equiv F_2^{\text{dat}}(x, Q^2) - 6 \frac{M^2 x^3}{Q^2 \tau^2} \int_{\xi}^1 \frac{dz}{z^2} F_2^{\text{dat}}(z, Q^2). \quad (4.38)$$

The integral over the experimentally measured structure function F_2^{dat} can be determined using the interpolation of the data based on neural networks which was constructed in Ref. [9]. Each pseudo-data point is then corrected using Eq. (4.38), and the fit proceeds as outlined in the previous sections, with F_2 replaced by Φ_2 .

Further higher-twist corrections are due to the contribution of subleading twist operators in the Wilson expansion. A possible next-to-leading twist contribution from these terms can be parametrized as

$$F_2^{\text{NLT}}(x, Q^2) = F_2^{\text{LT}}(x, Q^2) \left(1 + \frac{HT(x)}{Q^2} \right). \quad (4.39)$$

We will then assess the size of these corrections by parametrizing the function $HT(x)$ with a neural network and fitting it to the data, as we will discuss in Sect. 7.2.

x	$xu_v(x, Q_0^2)$ (LH)	$xu_v(x, Q_0^2)$ (NNPDF)	Rel. error
Leading order			
10^{-7}	$5.7722 \cdot 10^{-5}$	$5.7722 \cdot 10^{-5}$	$3.3760 \cdot 10^{-6}$
10^{-6}	$3.3373 \cdot 10^{-4}$	$3.3373 \cdot 10^{-4}$	$1.6880 \cdot 10^{-6}$
10^{-5}	$1.8724 \cdot 10^{-3}$	$1.8724 \cdot 10^{-3}$	$1.9212 \cdot 10^{-6}$
10^{-4}	$1.0057 \cdot 10^{-2}$	$1.0057 \cdot 10^{-2}$	$1.4095 \cdot 10^{-6}$
10^{-3}	$5.0392 \cdot 10^{-2}$	$5.0392 \cdot 10^{-2}$	$2.6145 \cdot 10^{-6}$
10^{-2}	$2.1955 \cdot 10^{-1}$	$2.1955 \cdot 10^{-1}$	$3.1065 \cdot 10^{-6}$
0.1	$5.7267 \cdot 10^{-1}$	$5.7267 \cdot 10^{-1}$	$6.4524 \cdot 10^{-6}$
0.3	$3.7925 \cdot 10^{-1}$	$3.7925 \cdot 10^{-1}$	$9.2674 \cdot 10^{-6}$
0.5	$1.3476 \cdot 10^{-1}$	$1.3476 \cdot 10^{-1}$	$1.1307 \cdot 10^{-5}$
0.7	$2.3123 \cdot 10^{-2}$	$2.3122 \cdot 10^{-2}$	$2.1165 \cdot 10^{-5}$
0.9	$4.3443 \cdot 10^{-4}$	$4.3440 \cdot 10^{-4}$	$6.3630 \cdot 10^{-5}$
Next-to-Leading order			
10^{-7}	$1.0616 \cdot 10^{-4}$	$1.0616 \cdot 10^{-4}$	$2.1462 \cdot 10^{-6}$
10^{-6}	$5.4177 \cdot 10^{-4}$	$5.4177 \cdot 10^{-4}$	$8.7799 \cdot 10^{-6}$
10^{-5}	$2.6870 \cdot 10^{-3}$	$2.6870 \cdot 10^{-3}$	$9.7796 \cdot 10^{-6}$
10^{-4}	$1.2841 \cdot 10^{-2}$	$1.2841 \cdot 10^{-2}$	$1.3380 \cdot 10^{-5}$
10^{-3}	$5.7926 \cdot 10^{-2}$	$5.7926 \cdot 10^{-2}$	$8.5063 \cdot 10^{-6}$
10^{-2}	$2.3026 \cdot 10^{-1}$	$2.3026 \cdot 10^{-1}$	$3.0757 \cdot 10^{-7}$
0.1	$5.5452 \cdot 10^{-1}$	$5.5452 \cdot 10^{-1}$	$7.6419 \cdot 10^{-7}$
0.3	$3.5393 \cdot 10^{-1}$	$3.5393 \cdot 10^{-1}$	$2.6979 \cdot 10^{-6}$
0.5	$1.2271 \cdot 10^{-1}$	$1.2271 \cdot 10^{-1}$	$2.4466 \cdot 10^{-5}$
0.7	$2.0429 \cdot 10^{-2}$	$2.0429 \cdot 10^{-2}$	$1.4810 \cdot 10^{-5}$
0.9	$3.6096 \cdot 10^{-4}$	$3.6094 \cdot 10^{-4}$	$6.0762 \cdot 10^{-5}$
Next-to-Next-to-Leading order			
10^{-7}	$1.5287 \cdot 10^{-4}$	$1.5287 \cdot 10^{-4}$	$1.5497 \cdot 10^{-5}$
10^{-6}	$6.9176 \cdot 10^{-4}$	$6.9176 \cdot 10^{-4}$	$5.0711 \cdot 10^{-6}$
10^{-5}	$3.0981 \cdot 10^{-3}$	$3.0981 \cdot 10^{-3}$	$9.5455 \cdot 10^{-6}$
10^{-4}	$1.3722 \cdot 10^{-2}$	$1.3722 \cdot 10^{-2}$	$1.8022 \cdot 10^{-5}$
10^{-3}	$5.9160 \cdot 10^{-2}$	$5.9160 \cdot 10^{-2}$	$5.0631 \cdot 10^{-6}$
10^{-2}	$2.3078 \cdot 10^{-1}$	$2.3078 \cdot 10^{-1}$	$2.4853 \cdot 10^{-6}$
0.1	$5.5177 \cdot 10^{-1}$	$5.5177 \cdot 10^{-1}$	$2.4747 \cdot 10^{-6}$
0.3	$3.5071 \cdot 10^{-1}$	$3.5071 \cdot 10^{-1}$	$2.8430 \cdot 10^{-7}$
0.5	$1.2117 \cdot 10^{-1}$	$1.2117 \cdot 10^{-1}$	$3.5893 \cdot 10^{-5}$
0.7	$2.0077 \cdot 10^{-2}$	$2.0077 \cdot 10^{-2}$	$5.5823 \cdot 10^{-6}$
0.9	$3.5111 \cdot 10^{-4}$	$3.5109 \cdot 10^{-4}$	$5.8172 \cdot 10^{-5}$

Table 3: Comparison of the Les Houches parton evolution benchmark results (LH) with our results (NNPDF).

N_{it}	$\langle \epsilon \rangle$	σ_ϵ	ϵ_{max}	# of points with $\epsilon \geq 5 \cdot 10^{-5}$.
1	$2.4 \cdot 10^{-2}$	$8.5 \cdot 10^{-2}$	$6.6 \cdot 10^{-1}$	21.3%
2	$1.3 \cdot 10^{-4}$	$6.2 \cdot 10^{-4}$	$4.0 \cdot 10^{-3}$	7.2%
4	$1.6 \cdot 10^{-5}$	$1.5 \cdot 10^{-4}$	$3.3 \cdot 10^{-3}$	5.4%
6	$1.6 \cdot 10^{-5}$	$1.5 \cdot 10^{-4}$	$3.3 \cdot 10^{-3}$	5.4%

Table 4: Relative accuracy ϵ (see text) in the result obtained for NLO evolution as a function of the number of points $N_{\text{pt}} = 16 (2^{N_{\text{it}}+1} - 1)$ used for gaussian integration.

x	$F_2^{\text{NS}}(x, Q^2)$ (Exact)	$F_2^{\text{NS}}(x, Q^2)$ (Interpolated)	Rel. error
Next-to-Next-to-Leading order			
10^{-3}	$3.7288 \cdot 10^{-2}$	$3.7288 \cdot 10^{-2}$	$1 \cdot 10^{-6}$
10^{-2}	$1.1750 \cdot 10^{-2}$	$1.1750 \cdot 10^{-2}$	$7 \cdot 10^{-6}$
0.1	$3.3497 \cdot 10^{-2}$	$3.3497 \cdot 10^{-2}$	$1 \cdot 10^{-5}$
0.3	$4.4425 \cdot 10^{-2}$	$4.4425 \cdot 10^{-2}$	$3 \cdot 10^{-7}$
0.5	$2.2159 \cdot 10^{-2}$	$2.2159 \cdot 10^{-2}$	$3 \cdot 10^{-5}$
0.7	$4.7996 \cdot 10^{-3}$	$4.7996 \cdot 10^{-2}$	$6 \cdot 10^{-5}$
0.9	$1.5031 \cdot 10^{-4}$	$1.5030 \cdot 10^{-4}$	$9 \cdot 10^{-5}$

Table 5: Comparison of the results for exact evolution at NNLO with coefficient function to results obtained interpolated evolution (see text). Evolution is performed from $Q_0^2 = 2 \text{ GeV}^2$ to $Q^2 = 230 \text{ GeV}^2$. The input is taken to be equal to the final central fit discussed in Sect. 6.

5. Neural networks

Neural networks, which we use in order to parametrize parton distributions, are nonlinear maps between input $\xi_i^{(1)}$ and output $\xi_i^{(L)}$ variables, i.e. they are just a particularly convenient set of interpolating functions. Like other sets of functions (such as e.g. orthogonal polynomials), neural networks, in the limit of infinite size, can reproduce any continuous function. However, the standard sets of basis function impose, upon truncation, a specific bias on the form of the fitted function: for example, polynomials of fixed degree can only have a maximum number of nodes and stationary points, and so on. This makes it difficult to obtain an unbiased representation of an unknown function.

Whereas clearly finite-size neural networks also impose a limitation on the set of functions that they can reproduce, their nonlinear nature makes it possible to make sure that this is not a source of bias. In particular, it is possible to show that stability of the fit is obtained, in that increasing the size of the neural network the results of the fit do not change. Very crudely speaking, this is a consequence of the fact that the smoothness of the fitted function decreases during the fitting process. As a consequence, it is possible to stop the fitting procedure before one attains the lowest value of the χ^2 (or other figure of merit), when the fit is optimized according to suitable statistical stopping criterion, but before statistical fluctuations are fitted. It is then possible to check that the result is independent of the size of the neural network: eventually, as the size increases, the best-fit result becomes independent of it. This is to be contrasted with standard fits, where one reaches the lowest χ^2 compatible with the given functional form, and eventually if one increases the size of the fitting function no stable fit can be obtained.

The critical aspects of the construction of a neural network parametrization are thus its form, its fitting (also called training) algorithm, and the criterion which is used to stop the training at the best fit. We now discuss in turn these points, by presenting both the general strategy and specific choices.

5.1 Structure

We use multi-layer feed-forward neural networks, schematically depicted in Fig. 6. In these neural networks, neurons are arranged into layers $l = 1, \dots, L$, with $j = 1, \dots, n_l$ neurons per layer. The output is given by neurons in the last layer, as a function of the output of all neurons in the previous layer, which in turn is a function of the output of all neurons in the previous layer and so on, starting from the first layer, which provides the input. The output $\xi_j^{(l)}$ of each neuron (j -th neuron of the l -th layer) is given by a nonlinear activation function $g(x)$:

$$\xi_i^{(l)} = g\left(h_i^{(l)}\right), \quad i = 1, \dots, n_l, \quad l = 2, \dots, L, \quad (5.1)$$

evaluated as a linear combination of the output $\xi_j^{(l-1)}$ of all neurons in the previous layers,

$$h_i^{(l)} = \sum_{j=1}^{n_{l-1}} \omega_{ij}^{(l)} \xi_j^{(l-1)} - \theta_i, \quad (5.2)$$

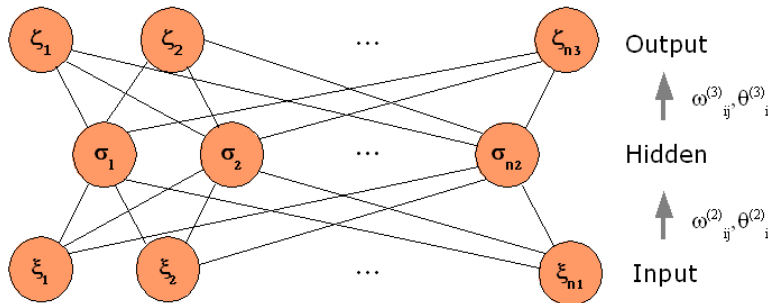


Figure 6: Schematic diagram of a feed-forward neural network.

where ω_{ij} (weights) and θ_i (thresholds) are free parameters to be determined by the fitting procedure, and $g(x)$ is taken to be a sigmoid in the inner layers,

$$g(x) = \frac{1}{1 + \exp(-x)}, \quad (5.3)$$

and linear $g(x) = x$ for the last layer. In practice, it turns out to be convenient to rescale both the input and the output of the neural network, in such a way that they both take values between 0 and 1: this avoids having weights ω_{ij} whose numerical values span many orders of magnitude. We have explicitly checked independence with respect to reasonable variations of this rescaling.

In short, the neural network outputs the values $\xi_j^{(l)}$ as a function of the input values $\xi_j^{(1)}$ and the parameters ω_{ij}, θ_i . The training of the neural network consists of the determination of the best-fit values of these parameters given a set of input-output patterns (data).

As discussed previously in detail [9], the choice of the architecture of the neural network cannot be derived from general rules and it must be tailored to each specific problem. One can roughly guess the size of the neural network based on rules of thumb, then verify by actual fitting which is the critical size above which results become independent of the size of the networks. Finally, one chooses a size which is somewhat larger than the critical one, in order to make sure that results are unbiased.

In previous work [9], it was found that a neural network with two hidden layers and architecture 4-5-3-1 is adequate for a fit of the nonsinglet structure function $F_2(x, Q^2)$. The neural network has four inputs because it turns out to be more efficient to take simultaneously as input both x, Q^2 , and their logs $\ln x, \ln Q^2$. In the present case, the neural network only fits the nonsinglet quark distribution at the initial scale, so only two input neurons for x and $\ln x$ are required. We then retain the same large architecture, which is surely redundant given that now only a function of a single variable is fitted. This leads us to the architecture 2-5-3-1. Stability of the results upon this choice will be tested in Sect. 6.3.

The neural network with this structure can be used to parametrize the nonsinglet quark distribution directly. However, it turns out to be convenient to actually relate the

neural network to the quark distribution through a suitable factor. This can be viewed as preprocessing of the data: if the function to be fitted is known to be dominated by some behavior in a given region, it may be convenient to only fit the deviation from this behavior.

In our case, it is known that the PDF has to satisfy the kinematical constraint $q_{\text{NS}}(x = 1, Q_0^2) = 0$. Standard counting rules [32] as well as existing parton fits further suggests that the structure function drops as a power as $x \rightarrow 1$ (up to logarithmic corrections): typically $q_{\text{NS}}(x = 1, Q_0^2) \underset{x \rightarrow 1}{\sim} (1 - x)^3$. This suggests that it might be useful to divide out this expected leading behavior as $x \rightarrow 1$. The vanishing constraint at $x = 1$ is then implemented automatically, which is more efficient than the solution adopted in previous work [9, 10], where the constraint was adopted by adding a Lagrange multiplier, i.e., in practice, artificial points at $F_2(x = 1, Q^2) = 0$ with various values of Q^2 .

Furthermore, the output of the neural network is by construction bounded. The last layer is a linear function of its input, which in turn has as its largest value the sum of weights, given that $\xi_j^{(n_l-1)}$ in turn are bounded, $\xi_j^{(n_l-1)} \leq 1$, according to Eq. (5.3). However, both current parton fits as well as Regge theory arguments [32] suggest that $q_{\text{NS}}(x, Q_0^2)$ blows up as $x \rightarrow 0$. Even though, of course, this growth is cut off by the fact that the data only extend down to a minimum value of x , and not to $x = 0$, this behavior leads to substantial variation of the PDF in the measured region as x decreases, and it is therefore advantageous to divide it out.

Therefore, we parametrize the nonsinglet parton distribution as

$$q_{\text{NS}}(x, Q_0^2) = \frac{(1 - x)^3}{x} \text{NN}(x) , \quad (5.4)$$

where $\text{NN}(x)$ is the neural network discussed above. Independence of the results on the various choices discussed in this section, in particular architecture and pre-processing, will be checked explicitly and discussed in the next section.

5.2 Training

The procedure of fitting (or training, as it is usually called in the context of neural networks) is based on the minimization of a suitable figure of merit. As discussed in Sect. 2, we fit to each replica of the data a nonsinglet quark distribution by maximum likelihood, i.e. we minimize the error function

$$E^{(k)}[\omega] = \frac{1}{N_{\text{dat}}} \sum_{i,j=1}^{N_{\text{dat}}} \left(F_i^{(\text{art})(k)} - F_i^{(\text{net})(k)} \right) \left((\overline{\text{cov}})^{-1} \right)_{ij} \left(F_j^{(\text{art})(k)} - F_j^{(\text{net})(k)} \right) , \quad (5.5)$$

where $F_i^{(\text{net})(k)}$ is determined in terms of the nonsinglet quark distribution at the reference scale Q_0^2 by Eq. (4.11), and the quark distribution at the reference scale is given by a neural network Eq. (5.4). The covariance matrix $\overline{\text{cov}}^{(k)}$ is defined in Eq. (3.7) and it does not include normalization errors, as discussed in Sect. 3.2. Note that $E^{(k)}$ is a property of each individual replica, whereas the quality of the global fit is given by the χ^2 computed from

the averages over the sample of trained neural networks, namely

$$\chi^2 = \frac{1}{N_{\text{dat}}} \sum_{i,j=1}^{N_{\text{dat}}} \left(F_i^{(\text{exp})} - \langle F_i^{(\text{net})} \rangle_{\text{rep}} \right) \left((\text{cov})^{-1} \right)_{ij} \left(F_j^{(\text{exp})} - \langle F_j^{(\text{net})} \rangle_{\text{rep}} \right), \quad (5.6)$$

where now the covariance matrix, defined in Eq.(3.1), includes normalization uncertainties.

The error function Eq. (5.5) can be minimized with a variety of techniques, including standard steepest-descent in the space of parameters. The back-propagation method, which was used in Refs. [9, 10] and is often advantageous for neural network training, cannot be adopted here because the measured quantity (the observed value of the structure function) depends non-locally on the neural network, i.e. it depends on the value of the parton distribution for several values of the input variable x . It turns out to be convenient to use a genetic algorithm [33] as a minimization method, as already done in Refs. [10, 11].

The genetic algorithm applied to our problem works in the following way, for each replica and repeating the whole procedure for each replica, i.e. N_{rep} times (we shall omit for simplicity the index k which identifies the individual replica). The state of the neural network is represented by the weight vector

$$\omega = (\omega_1, \omega_2, \dots, \omega_{N_{\text{par}}}) . \quad (5.7)$$

where each element ω_i corresponds to a weight $\omega_{ij}^{(l)}$ or threshold θ_i , and N_{par} indicates the total number of weights and thresholds. A set of N_{mut} copies of the state vector is then generated. Minimization is performed in steps usually referred to as generations (training cycles in ref. [10]). At each generation, a new set of copies of the weight vectors is obtained by means of two operations. First (mutation) a randomly chosen element of the state vector ω is replaced by a new value, according to the rule

$$\omega_k \rightarrow \omega_k + \eta \left(r - \frac{1}{2} \right) , \quad (5.8)$$

where r is a uniform random number between 0 and 1 and η (mutation rate) is a free parameter of the minimization algorithm which can be optimized either for the given problem, or dynamically during the minimization. Second (selection) a set of vectors with low values of the error Eq.(5.5) are selected out of the total population of N_{mut} individuals, and use to replace the original vector. The simplest option is to select the $N_{\text{sel}} < N_{\text{mut}}$ vectors with lowest error. Methods based on probabilistic selection, such as those used Ref. [10], explore more efficiently the space of possible mutations. Moreover, in order to avoid the local minima and increase the training speed, we have introduced multiple mutations. Specifically, we have found that one additional mutation with probability 50% and two additional mutations with probability 20% produce a significant improvement of the convergence rate.

The procedure is iterated until the vector with smallest value of the error function of the set for each generation meets a suitable convergence criterion, to be discussed in the next subsection. Note that at each generation the value of the error function never increases. The main advantage of genetic minimization is that it works on a population of solutions,

rather than tracing the progress of one point through parameter space. Thus, many regions of parameter space are explored simultaneously, thereby lowering the possibility of getting trapped in local minima. Usually, this procedure is further supplemented by crossing within the chains of mutated parameters (see e.g. Ref. [10]). However, this is not necessary in our case since, thanks to a dedicated preprocessing, mutations alone are enough to reach satisfactory minima. Moreover, the use of Monte Carlo replicas should avoid hitting the same minimum twice.

Here, genetic minimization is applied to an initial set of parameters chosen at random. In fact, in order to fully explore the space of parameters, it turns out to be advantageous to first choose at random the range of parameters, and then their values in this range. In practice, the range is chosen at random between

$$[-\langle\omega\rangle - \sigma_\omega, \langle\omega\rangle + \sigma_\omega] , \quad (5.9)$$

and

$$[-\langle\omega\rangle + \sigma_\omega, \langle\omega\rangle - \sigma_\omega] , \quad (5.10)$$

where $\langle\omega\rangle$ and σ_ω are the average and variance of the weights computed from the set of best-fit networks obtained in a previous fit. The value of the mutation rate is chosen to be $\eta = 8$, and the probabilistic algorithm of Ref. [9] is adopted. While these choices are optimized to obtain fast convergence, we have checked that they do not influence the final results.

Finally, because we have to deal with data sets coming from different experiments and with different features, it turns out to be advantageous to adopt weighted training, in order to ensure that the fit to all experiments is of comparable quality. To this purpose [10], more weight is given to experiments with a larger value of E . Namely, the function E_{\min} to be minimized is given by

$$E_{\min} = \frac{1}{N_{\text{dat}}} \sum_{j=1}^{N_{\text{exp}}} p_j N_{\text{dat},j} E_j , \quad (5.11)$$

where $N_{\text{dat},j}$ is the number of data points and E_j the value of the error function defined in Eq. (5.5) but restricted to the points coming from the j -th experiment, and the weights p_j are adjusted dynamically according to

$$p_j = \left(\frac{E_j}{E_{\max}} \right) , \quad (5.12)$$

where E_{\max} is the highest amongst the values of E_j . In order to avoid introducing artificial instabilities, the re-weighting is only used if the ratio on the right-hand side of Eq. (5.12) differs significantly from one. In practice, no reweighting is implemented if $c_{\min} < E_{\text{NMC}}/E_{\text{BCDMS}} < c_{\max}$, with $c_{\min} = 0.78$ and $c_{\max} = 1.22$.

5.3 Stopping

As we explained already, the crucial feature which guarantees a bias-free fit is the possibility of stopping the training not at the lowest value of the figure of merit (which might depend on

the functional form of the fitting function), but rather when suitable criteria are met. These criteria should single out the point where the fit reproduces the information contained in the data, but not the statistical noise. Namely, the best fit should have the lowest possible value of the figure of merit compatible with the requirement of not overlearning, i.e. not fitting statistical fluctuations. In previous work [9, 10], this was done by determining an optimal training length. The present case is more subtle, because on the one hand the fitted function is much more constrained by the data, on the other hand, the quantity which is being fitted is not directly observable. This makes it harder to distinguish overlearning from an actual improvement in the fit.

We therefore adopt the following criterion. We separate the data set into two disjoint sets. We then minimize the error function, Eq. (5.5) computed only with the data points of the first set (training set, henceforth). During the minimization process, we compute the error function from the data of the second set (validation set, henceforth). The best fit has been reached when the validation error function ceases to decrease. Overlearning, in particular, corresponds to a situation where the training error function keeps decreasing while the validation one does not, thereby signaling that the fit is merely reproducing the fluctuations of the specific training set. The procedure is reliable and it does not lead to loss of information if both the training and the validation set reproduce the features of the full data set. This can be simply achieved on average by choosing a different random partition for each replica of the data, and then checking that the number of replicas is sufficiently large. Similar methods have been widely used in various applications of neural networks, and in high-energy physics e.g. in Ref. [34].

In practice, we implement this stopping criterion as follows. We define a training fraction $f_{\text{tr}} = N_{\text{tr}}/N_{\text{dat}}$ (with default value $f_{\text{tr}} = 0.5$) and for each replica we select at random a fraction f_{tr} of points for each experiment, which we use for training, while the remaining points are assigned to a validation set. We then compute separately the corresponding training and validation covariance matrices. The training error function $E_{\text{tr}}^{(k)}$ Eq. (5.5) is minimized using a genetic algorithm, and at each generation of the genetic minimization the validation error function $E_{\text{val}}^{(k)}(l)$ is computed as a function of the generation index l . Having chosen a fixed fraction of points separately for each experiment allows for weighted training. Both $E_{\text{tr}}^{(k)}$ and $E_{\text{val}}^{(k)}(l)$ are computed neglecting cross-correlations between data in the validation and the training set: we have verified that this is a small correction anyway.

We then let the training proceed at least until $E_{\text{tr}}^{(k)}$ has reached the threshold value $E_{\text{tr}}^{(k)} = E_{\text{min}}$. We choose as default the value $E_{\text{min}} = 3$. This ensures that the training does not stop due to fluctuations in the early stages. Beyond this point, i.e. if $E_{\text{tr}}^{(k)} \leq E_{\text{min}}$, the training is stopped at the l -th generation if the training error function decreases,

$$\frac{\langle E_{\text{tr}}(l) \rangle}{\langle E_{\text{tr}}(l - N_{\text{sm}}) \rangle} < 1 \quad (5.13)$$

while the validation error function

$$\frac{\langle E_{\text{val}}^{(k)}(l) \rangle}{\langle E_{\text{val}}^{(k)}(l - N_{\text{sm}}) \rangle} \geq 1, \quad (5.14)$$

where

$$\langle E_{\text{val}}^{(k)} \rangle (l) \equiv \frac{1}{N_{\text{sm}}} \sum_{i=l-N_{\text{sm}}+1}^l E_{\text{val}}^{(k)}(i) , \quad (5.15)$$

with $E_{\text{val}}^{(k)}(i)$ the validation error function for the i -th GA generation, and similarly for the training error. In other words, the training is stopped if the value of the validation error function averaged over N_{sm} generation starts increasing, while the training error function decreases. This last condition must be imposed (despite the fact that with a genetic algorithm the figure of merit always decreases) because of weighted training: when the weighting is readjusted the error function could increase locally. We take $N_{\text{sm}} = 4$ as a default value. This averaging of the figures of merit along the training is analogous to the determination of moving averages, widely used especially in financial dynamics.

Finally, we have introduced an upper length to the training process, i.e. a maximum number of generation N_{gen} , chosen to correspond to a very long training such that the validation error function is no longer expected to decrease. The default value is $N_{\text{gen}} = 800$. In practice, the criterion condition (5.14) was always met well before this point except in a tiny number of cases.

6. Results

We present our best-fit result for the nonsinglet quark distribution and its statistical features, and in particular we show that it provides a consistent estimate of both central values and errors. We will then show its stability upon variation of the fitting procedure.

6.1 Next-to-leading order results: central values

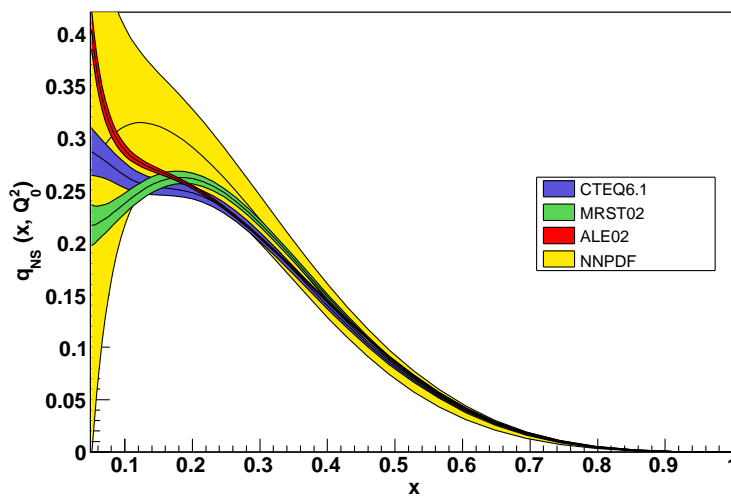


Figure 7: The best-fit NLO nonsinglet quark distribution $q_{\text{NS}}(x, Q_0^2)$ in the large- x region. The MRST, CTEQ and Alekhin determinations are also shown for comparison. In this and subsequent plots of q_{NS} we take $Q^2 = Q_0^2 = 2 \text{ GeV}^2$.

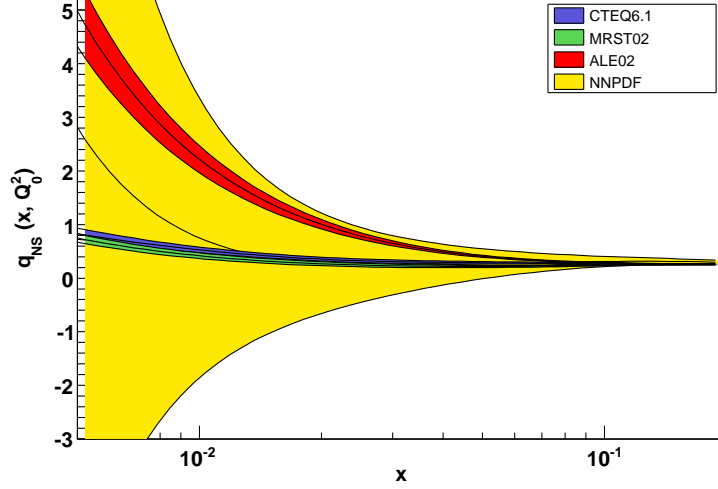


Figure 8: The best-fit NLO nonsinglet quark distribution $q_{\text{NS}}(x, Q_0^2)$ in the small- x region. The MRST, CTEQ and Alekhin determinations are also shown for comparison.

The result obtained from a set of $N_{\text{rep}} = 1000$ neural networks trained to replicas of the data with the method described above, using the NLO expressions of the coefficient function and anomalous dimension is displayed in Figs. 7 and 8. The error band corresponds to a one-sigma contour. The distributions of values of the error function E Eq. (5.5) for the training and validation samples are displayed in Fig. 9. These distributions are poissonian (gaussian) to good approximation, and both are peaked around a similar value, thereby showing that the quality of the fit for points included or not included in the fit is equally good. The distribution of training lengths is shown in Fig. 10. It is poissonian to good approximation, and it shows that convergence is reached after a small number of generations of the genetic algorithm, much smaller than the maximum N_{gen} which is only reached in a tiny number of cases.

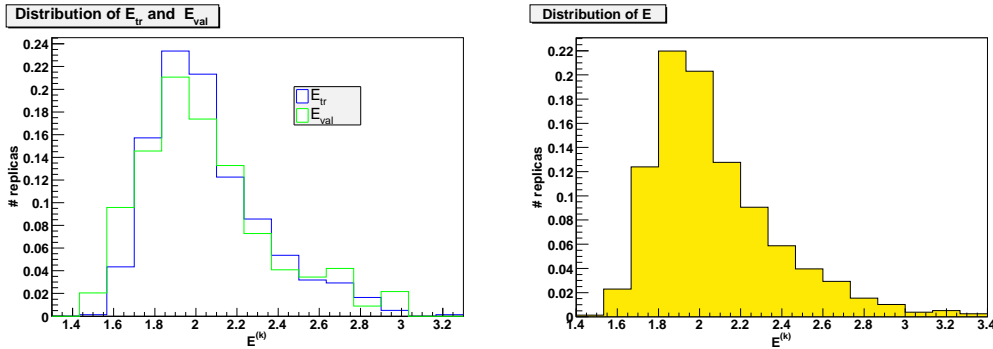


Figure 9: Distribution of values of the error function E Eq. (5.5) at the stopping point for the training and validation samples (left), and for the full data sample (right).

The distribution of values of the error function E Eq. (5.5) for the total data set is displayed in Fig. 9, while the statistical properties of the fit are summarized in Table 6. Note that, because the quantities listed in the table characterize the comparison of the fit to the data (see Appendix B for the definitions), they are determined from the structure function F_2^{NS} rather than the quark distribution itself. The error functions of each fit are distributed to good approximation in a gaussian way about the value $\langle E \rangle = 2.27$ given in Table 6. A value $\langle E \rangle \approx 2$ is expected for a good fit because if errors are correctly estimated, the average standard deviation of true values about the measured value should equal the total uncertainty, but replicas are generated about the measured values so their average standard deviation about the true values should be twice the total uncertainty. However, the fit to each individual replica should be closer to the true value, because of the availability of many data points at various values of Q^2 which are combined by the fit. Indeed, the average error of the fit is much smaller than the average experimental error: $\langle \sigma^{(\text{net})} \rangle_{\text{dat}} \ll \langle \sigma^{(\text{exp})} \rangle_{\text{dat}}$ (see Table 6).

The fact that the figure of merit for the global fit Eq. (5.6) takes the value $\chi^2 \approx 1$ shows that the central value of the fit, after averaging over replicas, is distributed about the measured values as one expects of the correct distribution of true values, namely, with variance equal to the experimental error (which is much larger than the error on the fit itself as we just discussed). Note that, strictly speaking, the total number of independent degrees of freedom is somewhat smaller than the total number of points. However, the number of effective free parameters of the neural network is rather smaller than the total number of weights and thresholds, because our choice of stopping criterion ensures that the neural network is redundant. Hence, the number of degrees of freedom is at most by a few percent smaller than N_{dat} (and the correctly estimated χ^2 accordingly larger), which is smaller than the uncertainty on χ^2 due to statistical fluctuations, which is of order of $N_{\text{dat}}^{-1/2}$. However, as a consequence of this, we expect the value of χ^2/N_{dat} to be systematically lower than one by a few percent for an optimal fit. This expectation will be borne out by our results.

We conclude that the central fit is correctly estimated.

6.2 Next-to-leading order results: uncertainties

Because experimental errors are much larger than the error on the fit, the values of $\langle E \rangle$ and χ^2 discussed in Sect. 6.1 do not test for the accuracy of the one-sigma error band on the central fit. We can check for the correctness of the latter by studying the fluctuation of predictions obtained from different sets of replicas. To this purpose, we note that we can view the prediction q_i obtained for the central values of the quark distribution by averaging

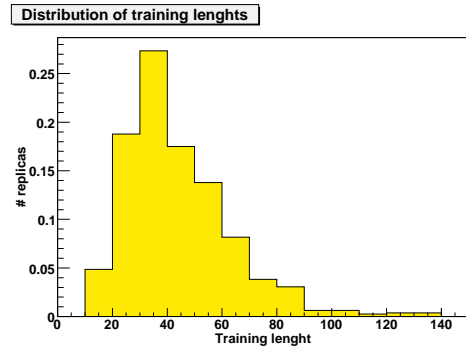


Figure 10: Distribution of training lengths.

	Total	NMC	BCDMS
χ_{tot}^2	0.75	0.72	0.78
$\langle E \rangle$	2.27	1.99	2.52
$r [F_2^{\text{NS}}]$	0.81	0.66	0.95
$\langle \sigma^{(\text{exp})} \rangle_{\text{dat}}$	0.011	0.017	0.006
$\langle \sigma^{(\text{net})} \rangle_{\text{dat}}$	0.006	0.009	0.004
$r [\sigma^{(\text{net})}]_{\text{dat}}$	0.59	-0.04	0.86
$\langle \rho^{(\text{exp})} \rangle_{\text{dat}}$	0.11	0.39	0.16
$\langle \rho^{(\text{net})} \rangle_{\text{dat}}$	0.46	0.42	0.50
$r [\rho^{(\text{net})}]_{\text{dat}}$	0.15	0.25	0.04
$\langle \text{cov}^{(\text{exp})} \rangle_{\text{dat}}$	$8.6 \cdot 10^{-6}$	$1.0 \cdot 10^{-5}$	$7.2 \cdot 10^{-6}$
$\langle \text{cov}^{(\text{net})} \rangle_{\text{dat}}$	$2.1 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$	$6.9 \cdot 10^{-6}$
$r [\text{cov}^{(\text{net})}]_{\text{dat}}$	0.24	0.23	0.57

Table 6: Statistical estimators for the final sample of $N_{\text{rep}} = 1000$ neural networks, both for the total data points and for the individual experiments incorporated in the fit (NMC and BCDMS).

over N_{rep} replicas, as a random variable, whose variance is given by

$$V[q_i] = \frac{\sigma_{q_i}^2}{N_{\text{rep}}}, \quad (6.1)$$

in terms of the variance $\sigma_{q_i}^2$ of the set of replicas. We can define an average distance $d[q]$ as the root-mean square difference between values of q_i obtained by averaging over different set of replicas, normalized to the variance Eq. (6.1) (see appendix B). If the error on q is correctly estimated, the average of $d[q]$, determined for different points, must approach one. We can similarly view the standard deviation of each data point $\sigma_{q_i}^2$, and test for their statistical accuracy by computing the corresponding distances.

Hence, we test for accuracy of estimates of errors and correlations by computing this distance for the quark distribution q_i (determined as an average over neural networks) and the error on it $\sigma_{q_i}^2$ (determined as variance of the networks). The distance is determined between results obtained from a set of N_{rep} replicas each. The computation is performed for 14 points linearly spaced in x in the data region ($0.05 \leq x \leq 0.75$) and then averaged, and similarly for the same number of points logarithmically spaced in the extrapolation region ($10^{-3} \leq x \leq 10^{-2}$). In order for the result to be significant we must make sure that its standard deviation σ_d is not too large. However, because d is a standard deviation itself (normalized to its expected value), σ_d only scales as $n^{-1/4}$ with the number of points n , so $\sigma_d \sim 0.5$. On the other hand, increasing the number of points will not help because, of course, two values of $q(x)$ and $q(x')$ are highly correlated when x and x' are close to each other. Thus, we obtain stability by repeating the

N_{rep}	10	100	500
$\langle d[q] \rangle_{\text{dat}}$	1.02	0.96	0.92
$\langle d[q] \rangle_{\text{extra}}$	1.11	0.99	0.85
$\langle d[\sigma_q] \rangle_{\text{dat}}$	0.93	0.88	0.93
$\langle d[\sigma_q] \rangle_{\text{extra}}$	1.13	0.97	0.91

Table 7: Stability estimators as a function of the number of trained replicas N_{rep} .

computation for different random choices of two sets of N_{rep} replicas among the starting $2N_{\text{rep}}$ ones, and averaging the results, which reduces the standard deviation according to Eq. (6.1). We average over 1000 sets, so that $\sigma_d \sim 0.01$.

Results with increasing values of N_{rep} are shown in Table 7. The distance already converges to the expected value of one, to percent accuracy, when $N_{\text{rep}} = 10$. Note that, because of Eq. 6.1, the expected distance decreases as $\frac{1}{\sqrt{N_{\text{rep}}}}$: the stability of the result we get when the number of replicas is further increased from 100 to 500 shows that this behaviour is indeed observed in our sample of neural networks.

6.3 Stability upon variation of the neural network structure

After verifying that our results are statistically consistent, we wish to check that they do not depend on the details of the fitting procedure. We do this by determining the distance discussed in Sect. 6.2, but now computed for a pair of fits which differ in fitting procedure. Independence is verified if the distance equals its value predicted on statistical grounds (i.e. that which would be obtained from pairs of replicas coming from the same fit). Specifically, the computation is done by taking $N_{\text{rep}} = 50$, and then averaging over 1000 different choices of these 50 replicas among a total set of $N_{\text{rep}} = 100$ replicas. Note that, because of Eq. 6.1, this verifies independence of the fitting procedure to an accuracy which is by a factor $\sqrt{50}$ smaller than the uncertainty on the central value.

The first check is to verify that our results are indeed independent of the neural network architecture. We have compared the results of the fit with the reference architecture 2-5-3-1 with the results of a similar fit with a smaller neural network architecture 2-4-3-1. The result of this comparison can be seen in Table 8. This confirms that we have achieved independence of the number of parameters used, both in the data and in the extrapolation region, a property that is very difficult to achieve in standard parton fits with fixed functional form.

Architecture	2-4-3-1
χ^2	0.75
$\langle d[q] \rangle_{\text{dat}}$	0.9
$\langle d[q] \rangle_{\text{extra}}$	0.9
$\langle d[\sigma_q] \rangle_{\text{dat}}$	0.9
$\langle d[\sigma_q] \rangle_{\text{extra}}$	1.4

Table 8: Independence of the architecture of the neural networks.

Next, we estimate the dependence of the results on the different kind of preprocessing, namely, on the values of the exponents m and n

$$q_{\text{NS}}(x, Q_0^2) = \frac{(1-x)^m}{x^n} NN(x), \quad (6.2)$$

when varied about the default values $m = 3$ and $n = 1$. In Table 9 we display the dependence of the χ^2 , computed from a set of $N_{\text{rep}} = 100$ replicas, and the stability of the fit when $1 \leq m \leq 4$ (with $n = 1$ fixed) and when $0.5 \leq n \leq 1.25$ (with $m = 3$ fixed). It appears that the fit is reasonably stable if $2.5 \leq m \leq 3.5$ and $0.5 \leq n \leq 1.25$: central values in the data region fluctuate less than three σ while the uncertainty is stable. The central fit starts changing in a statistically significant way when m and n are varied outside these bounds. However, in such case the quality of the fit, as measured by the χ^2 , deteriorates considerably. Hence, we conclude that the fit is independent of the choice of m, n provided these are kept within a range which allows for an optimal quality of the fit.

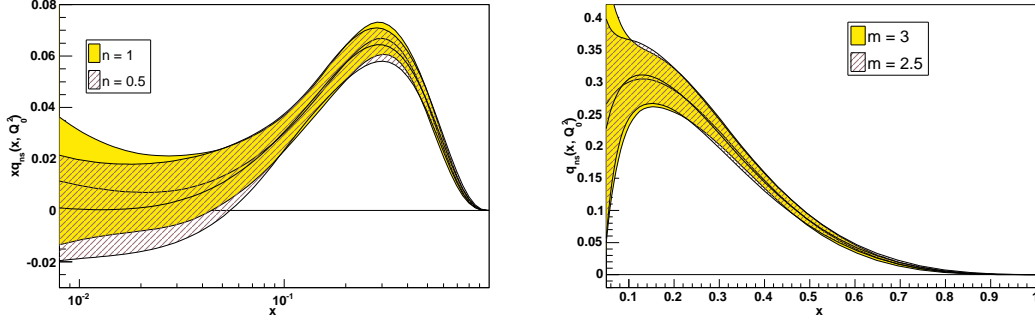


Figure 11: Dependence of the results on the preprocessing exponents m, n , Eq. (6.2)

(m, n)	(1,1)	(2,1)	(2.5,1)	(3,1)	(3.5,1)	(4,1)
χ^2	0.85	0.94	0.83	0.75	0.77	0.84
$\langle d[q] \rangle_{\text{dat}}$	4.5	4.6	2.9	-	3.3	6.1
$\langle d[q] \rangle_{\text{extra}}$	0.8	0.8	0.8	-	0.8	0.8
$\langle d[\sigma_q] \rangle_{\text{dat}}$	2.0	1.7	1.2	-	1.2	1.5
$\langle d[\sigma_q] \rangle_{\text{extra}}$	1.9	2.0	1.3	-	1.0	1.1

(m, n)	(3,0.5)	(3,0.75)	(3,0.9)	(3,1)	(3,1.1)	(3,1.25)
χ^2	0.79	0.77	0.74	0.75	0.76	0.77
$\langle d[q] \rangle_{\text{dat}}$	2.5	2.1	1.0	-	1.2	1.6
$\langle d[q] \rangle_{\text{extra}}$	1.2	0.8	0.9	-	1.9	1.5
$\langle d[\sigma_q] \rangle_{\text{dat}}$	1.3	1.3	0.9	-	1.0	1.3
$\langle d[\sigma_q] \rangle_{\text{extra}}$	1.0	1.3	1.2	-	1.8	2.5

Table 9: Dependence on the preprocessing exponents.

Fits within the stability region of the preprocessing exponents are compared in Fig. 11. Stability of central values and errors in the data and extrapolation region are apparent. Note that the distance Eq. (6.1) is computed with $N_{\text{rep}} = 50$, hence the distance of the central values equals about 15% of the error band displayed in the plot.

7. Theoretical uncertainties

So far we have checked that our fit reproduces correctly the information contained in the data, and thus that the error band computed from it is statistically consistent. A priori, there are further sources of theoretical error related to use of perturbative QCD in the fitting procedure: specifically those related to higher order perturbative corrections and to higher twist terms. In this section we will show that these theoretical errors are in fact negligible. All tests in this section, as in Sect. 6.3 are done by computing the χ^2 from a set of 100 replicas, and the distance by taking $N_{\text{rep}} = 50$, and then averaging over 1000 different choices of these 50 replicas among a total set of $N_{\text{rep}} = 100$ replicas.

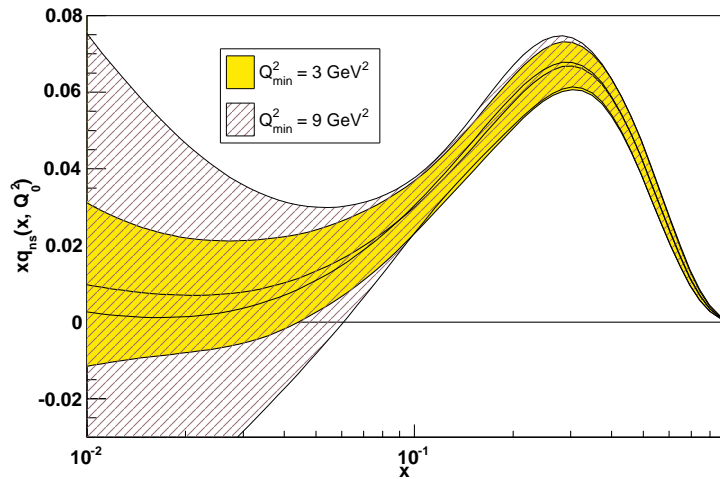


Figure 12: Results for $xq_{\text{NS}}(x, Q_0^2)$ with two different kinematical cuts in Q^2 .

7.1 Dependence on kinematic cuts

We first discuss the dependence of results for the nonsinglet parton distribution $q_{\text{NS}}(x, Q^2)$ on the kinematical cut in Q^2 . In Fig. 12 and Table 10 we compare results obtained when the value Q_{min}^2 is raised to $Q_{\text{min}}^2 = 9 \text{ GeV}^2$, to results obtained with the default value $Q_{\text{min}}^2 = 3 \text{ GeV}^2$. Removing the data points in the region $3 \text{ GeV}^2 \leq Q^2 \leq 9 \text{ GeV}^2$ eliminates the bulk of the NMC measurements, i.e. (see Fig. 2) essentially all data with $x < 0.1$.

As a consequence, uncertainties increase sizably in the small- x region. However, results are remarkably stable in the large x region. In Tab. 10 we compare results in three different kinematical regions: where the two fits with different kinematical cuts have data (region I, $0.2 \leq x \leq 0.75$), where only the fit with $Q_{\text{min}}^2 = 3 \text{ GeV}^2$ has data (region II, $0.05 \leq x \leq 0.1$) and finally the region where both fits extrapolate (region III, $x \leq 10^{-2}$). Not only in region I the two fits agree completely, but even in region II, whereas the uncertainty bands increases considerably, the central fit is essentially unaffected. This proves the remarkable stability of our results.

	$x \leq 10^{-2}$	$5 \cdot 10^{-2} \leq x \leq 0.1$	$0.2 \leq x \leq 0.75$
$\langle d[q] \rangle$	0.6	0.8	0.9
$\langle d[\sigma] \rangle$	2.7	3.2	1.4

Table 10: Comparison of fits with different kinematical cuts. The fit with $Q_{\text{min}}^2 = 3 \text{ GeV}^2$ (reference) is compared to the fit with $Q_{\text{min}}^2 = 9 \text{ GeV}^2$ in three different regions of x .

7.2 Higher twists

We now test for the possible role of higher twist terms, on top of the target-mass corrections which are already included in our default fit as discussed in Sect. 4.4. To this purpose, we switch on a twist-four contribution according to Eq. (4.39), and parametrize $HT(x)$

Fit	$Q_{\min}^2 = 3 \text{ GeV}^2 + \text{HT}$	$Q_{\min}^2 = 5 \text{ GeV}^2$	$Q_{\min}^2 = 5 \text{ GeV}^2 + \text{HT}$
χ^2	0.76	0.79	0.78
$\langle d[q] \rangle_{\text{dat}}$	2.9	0.8	3.2
$\langle d[q] \rangle_{\text{extra}}$	1.4	0.8	0.9
$\langle d[\sigma_q] \rangle_{\text{dat}}$	1.2	1.5	1.9
$\langle d[\sigma_q] \rangle_{\text{extra}}$	1.3	1.8	2.3

Table 11: Results of the higher-twist analysis. Stability is computed relative to the default NLO fit.

with a 1-2-1 neural network. In Table 11 we compare the χ^2 obtained by including this higher twist term to that obtained by raising the cut in Q^2 . On the one hand, we see that including higher-twist corrections without raising the Q^2 does not lead to any improvement in the quality of the fit. On the other hand, we see that if the Q^2 is raised, the χ^2 also does not improve — in fact it deteriorates slightly, in a way which is unaffected by the inclusion of higher twist terms.

We take the lack of improvement in χ^2 when either the cut is raised, or higher twist terms are included, as an indication of the fact that there is no evidence whatsoever for higher twist terms even with our lower default cut. The slight deterioration in χ^2 when the Q^2 cut is raised is likely to be related to the fact that experimental uncertainties in the small- Q^2 region (i.e. experimental uncertainties on NMC data) are somewhat overestimated.

We conclude that there is no evidence for higher twists for nonsinglet F_2^{NS} data with $Q^2 > 3 \text{ GeV}^2$.

7.3 Higher orders and the value of the strong coupling

So far we have discussed the determination of the nonsinglet parton distribution through a NLO analysis. We now compare these results to those obtained at one less and one more perturbative order. Hence, we produce a full LO fit, with $\alpha_s(M_Z^2) = 0.130$ and a full NNLO fit, with $\alpha_s(M_Z^2) = 0.115$. The results of these fits are compared to the NLO ones in Table 12. No variation of the quality of the fit is found between different perturbative orders: the values of χ^2 and $\langle \sigma^{(\text{net})} \rangle_{\text{dat}}$ are unaffected by the perturbative order at which the computation is performed. Indeed, only very small improvements of the χ^2 when going from LO to NLO and NNLO have been found in recent nonsinglet fits [35, 36].

This means that the effects of higher-order corrections are entirely reabsorbed in a change of the initial quark distribution. This change is statistically significant when going from LO to NLO: the central values moves by $d \approx 10$ (see Tab. 12), i.e. by about half the error on the central value. Indeed, we have verified that if Tab. 12 is recomputed using a set of $N_{\text{rep}} = 50$ replicas, we get $\langle d[q] \rangle_{\text{dat}} = 3.2$ for the LO fit. This means that the LO and NLO central values differ by a fixed amount, independent of the number of replicas used to determine the central value, therefore if we normalize to the standard deviation Eq. 6.1 the result grows as $\sqrt{N_{\text{rep}}}$. No statistically significant difference in central values is observed when going from NLO to NNLO. Hence, our result supports the

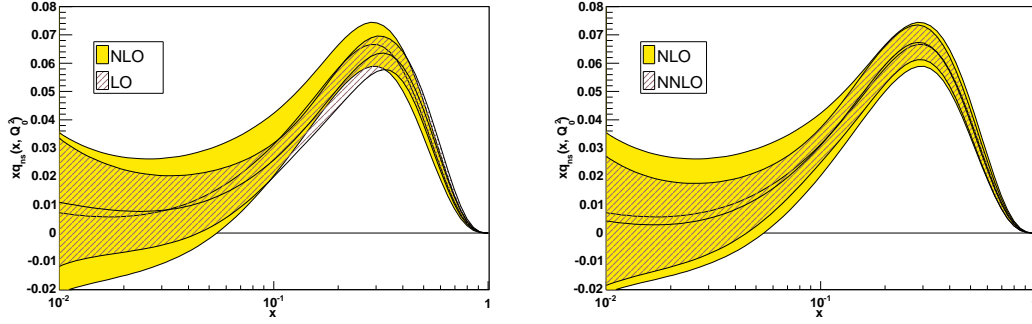


Figure 13: Comparison of a LO fit (left) and a NNLO fit (right) to the reference NLO fit.

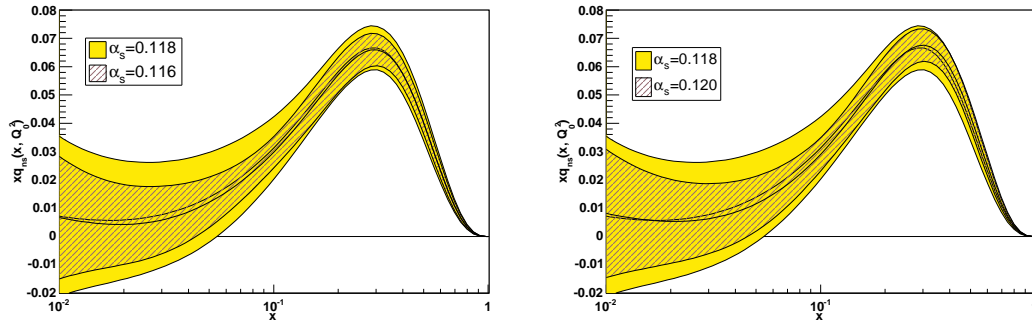


Figure 14: Comparison of the reference fit to fits with a lower (left) or higher (right) value of $\alpha_s(M_Z^2)$

conclusion [35] that nonsinglet data are insufficient to provide evidence for higher-order perturbative corrections. These results are apparent in Fig. 13, where the LO, NLO and NNLO fits are compared.

Perturbative order	LO	NLO	NNLO
χ^2	0.751	0.750	0.754
$\langle d[q] \rangle_{\text{dat}}$	10.2	-	2.6
$\langle d[q] \rangle_{\text{extra}}$	1.2	-	1.2
$\langle d[\sigma_q] \rangle_{\text{dat}}$	2.2	-	3.0
$\langle d[\sigma_q] \rangle_{\text{extra}}$	1.3	-	1.3
$\langle \sigma^{(\text{net})} \rangle_{\text{dat}}$	$0.51 \cdot 10^{-2}$	$0.62 \cdot 10^{-2}$	$0.49 \cdot 10^{-2}$

Table 12: Quantitative comparison of fits at different perturbative orders. The distance is computed relative to the NLO fit using a set of $N_{\text{rep}} = 500$ replicas.

The fact that NLO corrections have a negligible impact on the fit suggests that the fit is only weakly dependent on the value of $\alpha_s(M_Z^2)$. We have repeated our NLO fit while varying the value of $\alpha_s(M_Z^2)$ by one sigma about its central value, i.e. with $\alpha_s(M_Z^2) = 0.116$ and $\alpha_s(M_Z^2) = 0.120$. Comparison to the central fit (Fig. 14 and Tab. 13) shows that the variation of the central fit due to the change in $\alpha_s(M_Z^2)$ in this range has marginal

$\alpha_s(M_Z^2)$	0.116	0.118	0.120
χ^2	0.743	0.750	0.744
$\langle d[q] \rangle_{\text{dat}}$	3.8	-	4.1
$\langle d[q] \rangle_{\text{extra}}$	0.8	-	0.7
$\langle d[\sigma_q] \rangle_{\text{dat}}$	1.6	-	2.4
$\langle d[\sigma_q] \rangle_{\text{extra}}$	1.4	-	1.5
$\langle \sigma^{(\text{net})} \rangle_{\text{dat}}$	$0.52 \cdot 10^{-2}$	$0.62 \cdot 10^{-2}$	$0.53 \cdot 10^{-2}$

Table 13: Quantitative comparison of fits at NLO with different values of $\alpha_s(M_Z^2)$.

statistical significance: $\langle d[q] \rangle_{\text{dat}} \sqrt{N_{\text{dat}}} \approx 0.2$, i.e., the central fit moves by about a fifth of a standard deviation when $\alpha_s(M_Z^2)$ is varied in this range. This variation, though small, appears to be stable: if 50 replicas are used we get $\langle d[q] \rangle_{\text{dat}} = 1.3$ for $\alpha_s(M_Z^2) = 0.116$ and $\langle d[q] \rangle_{\text{dat}} = 1.1$ for $\alpha_s(M_Z^2) = 0.120$, corresponding to the same value of $\langle d[q] \rangle_{\text{dat}} \sqrt{N_{\text{dat}}} \approx 0.2$. The variation of the χ^2 in this range is also marginal and it does not show a clear trend. We conclude that a NLO determination of α_s based on nonsinglet data only is in principle possible, but it is necessarily affected by an uncertainty which is sizably larger than the error 0.002 on the PDG average. A smaller value for this uncertainty would be a likely indication of an underestimate of the uncertainty related to the choice of parton parametrization.

8. The structure function

We finally turn to results for the physically measurable structure function $F_2^{\text{NS}}(x, Q^2)$, which we compare to the data and to results obtained in various other approaches.

8.1 Comparison to data and other parton fits

In Figs. 15 and 16 we show a comparison of our results for the nonsinglet structure function $F_2^{\text{NS}}(x, Q^2)$ as a function of x at fixed $Q^2 = 15 \text{ GeV}^2$ compared to all data with $13 \leq Q^2 \leq 17 \text{ GeV}^2$, and as a function of Q^2 at fixed $x = 0.15$, compared to all data in the range $0.13 \leq x \leq 0.17$. In the same figure, we also show the structure function obtained from the results of the global fits of the CTEQ [5] and MRST [6] collaborations and the fit to deep-inelastic data by Alekhin [4].

The uncertainty in the final structure function is much smaller than that on the data, thanks to the theoretical information from perturbative evolution. Nevertheless, the uncertainty in our determination, especially at small x , is rather larger than that found in previous analyses. One may wonder whether this is due to the fact that the global analyses are based on a wider set of data. However, in the small x region, where the difference between the uncertainty on our result and that found by other groups is most striking, none of the data contained in these global fits can constrain $q_{\text{NS}}(x, Q_0^2)$, on top of the data on $F_2^{\text{NS}}(x, Q^2)$ which we also include. This suggests that the smaller error band obtained by these groups may be due to parametrization bias or uncertainty underestimation.

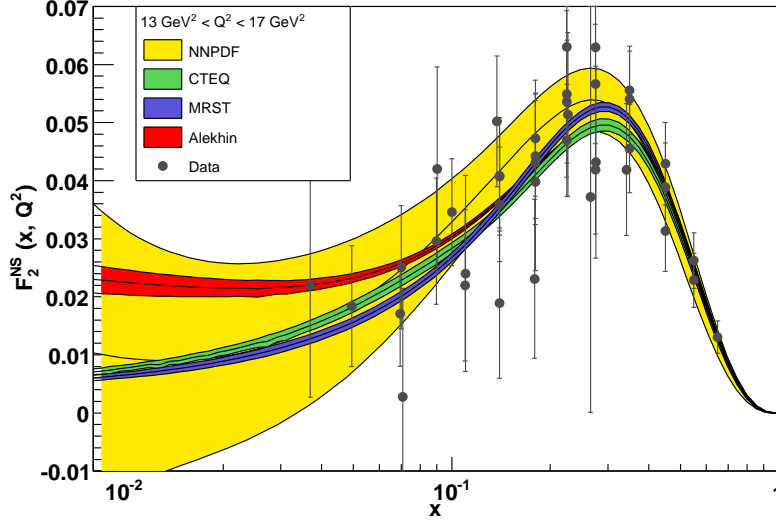


Figure 15: Our NLO determination of the nonsinglet structure function $F_2^{\text{NS}}(x, Q^2)$ at fixed $Q^2 = 15 \text{ GeV}^2$ compared to all data with $13 \leq Q^2 \leq 17 \text{ GeV}^2$. Determinations by other groups are also shown.

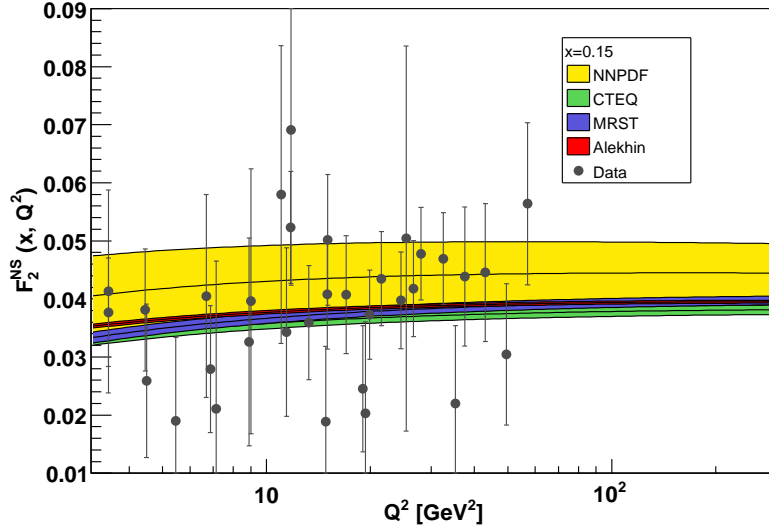


Figure 16: Our NLO determination of the nonsinglet structure function $F_2^{\text{NS}}(x, Q^2)$ as a function of Q^2 for $x = 0.15$, compared to all data in the range $0.13 \leq x \leq 0.17$. Determinations by other groups are also shown.

Be that as it may, whereas all available fits are within our error band, our central result disagrees with these fits, which in turn disagree with each other within respective errors, even in the valence region $0.1 \leq x \leq 0.3$.

Finally, it is interesting to compare our NNLO determination of the nonsinglet parton distribution to the results of some recent NNLO fits to nonsinglet DIS data [35, 36] (see

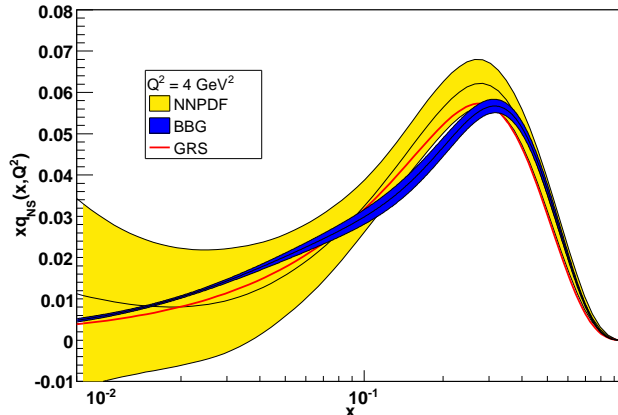


Figure 17: Comparison of our NNLO fit with the results of the nonsinglet fits of Refs. [36] and [35].

Fig. 17). Results are analogous to those found when comparing to global fits.

8.2 Comparison to neural $F_2^{\text{NS}}(x, Q^2)$

In Ref. [9] the nonsinglet structure function $F_2^{\text{NS}}(x, Q^2)$ was directly parametrized with neural networks as a function of its two arguments, instead of being obtained by evolving a parton distribution. The results of that analysis are compared to the present NLO determination in Fig. 18. The comparison shows that the uncertainty on the determination of Ref. [9] is already smaller than that of the data, because of the requirement of smoothness of the function which is fitted to them, as discussed in detail in Ref. [9]. However, thanks to the extra information from perturbative evolution, the uncertainty is further reduced by a considerable amount when fitting the structure function. Also, it is apparent that thanks to this error reduction, a more detailed determination of the shape of F_2^{NS} is possible. We also display the pull (see Appendix B) between these two determinations, which shows that they are in perfect agreement within one sigma in the region where there are data: whenever the pull increases in modulo above one, no data are available.

9. Conclusions and outlook

In this paper we have provided a first determination of a parton distribution within an approach based on the use of neural networks coupled to a Monte Carlo representation of the probability density in the space of structure functions.

Our final results take the form of a Monte Carlo sample of neural networks. The nonsinglet parton distribution $q_{\text{NS}}(x, Q_0^2)$ and its statistical moments (errors, correlations and so on) can be determined by averaging over this sample. These results together with driver FORTRAN code can be downloaded from the web site <http://sophia.ecm.ub.es/nnpdf/>.

The main distinctive features of our determination of the isotriplet quark distribution when compared to that obtained in other fits are the following:

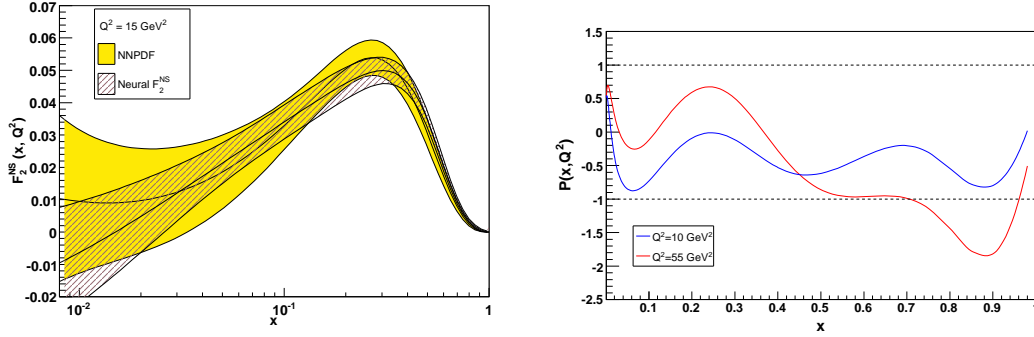


Figure 18: Comparison of the structure function $F_2^{\text{NS}}(x, Q^2)$ obtained from our NLO $q_{\text{NS}}(x, Q_0^2)$ with the NNPDF parametrization of $F_2^{\text{NS}}(x, Q^2)$ at $Q^2 = 15 \text{ GeV}^2$: structure function (left) and pull (right). The pull at $Q^2 = 55 \text{ GeV}^2$ is also shown (lower curve at large x).

- Our fit is demonstrably independent of the parametrization, and the uncertainty on it has the correct statistical behavior which one expects of an unbiased statistical estimator.
- The uncertainty is greatly reduced in comparison to that on the data, and also to that obtained in fits [9] which do not exploit the constraint of perturbative parton evolution.
- The error band on our result, especially at small x , is rather larger than that of other fits, and in fact it includes these other fits even when they disagree with each other.
- The quality of fits at LO, NLO and NNLO is the same: difference in perturbative order can be reabsorbed in the choice of boundary condition.

Our approach is designed to tackle some of the most serious problems that plague current global parton fits, namely, the difficulty of keeping under control the bias due to the choice of parametrization, problems in combining different data sets, and the problem of determining faithfully the uncertainty on the final result. This first application to the determination of a single parton distribution from all structure function data shows that the approach is viable and successful. Its ultimate success or failure will be in its application to a global parton fit from all available data.

Acknowledgments

This work was partly supported by grants ANR-05-JCJC-0046-01 (France), PRIN-2004 (Italy), MEC FIS2004-05639-C02-01 and AP2002-2415 (Spain). We thank R. D. Ball, M. Cacciari, G. d'Agostini, A. Guffanti, L. Magnea, G. Ridolfi and G. Salam for discussions during the course of this work.

A. Large x behavior of the evolution factor

At leading order in $\alpha_s(Q^2)$, in the large x limit, the dominant contribution to the evolution kernel comes from the large N limit of the LO anomalous dimension, given by

$$\Gamma(x) = \int_{-i\infty}^{+i\infty} \frac{dN}{2\pi i} x^{-N} \left(\frac{\alpha_s(Q^2)}{\alpha_s(Q_0^2)} \right)^{C_F \ln N / \pi \beta_0} . \quad (\text{A.1})$$

The inverse Mellin transformation can be performed to all logarithmic orders using the formulas of Ref. [37], with the result

$$\begin{aligned} \Gamma(x) &= \sum_{n=1}^{\infty} \frac{\Delta^{(n-1)}(1)}{(n-1)!} \left[\frac{1}{1-x} \frac{d^n}{d \ln^n(1-x)} \left(\frac{\alpha_s(Q^2)}{\alpha_s(Q_0^2)} \right)^{C_F \ln(1-x) / \pi \beta_0} \right]_+ + \mathcal{O}((1-x)^0) \\ &= \Delta \left(\frac{C_F}{\pi \beta_0} \ln \left(\frac{\alpha_s(Q_0^2)}{\alpha_s(Q^2)} \right) \right) \left[(1-x)^{-1 + \frac{C_F}{\beta_0 \pi} \ln \left(\frac{\alpha_s(Q_0^2)}{\alpha_s(Q^2)} \right)} \right]_+ + \mathcal{O}((1-x)^0) . \end{aligned} \quad (\text{A.2})$$

where $\Delta(\eta) = 1/\Gamma(\eta)$. If $Q^2 > Q_0^2$ the exponent of $1-x$ in Eq. (A.2) is larger than one, thereby ensuring that, thanks to the $+$ prescription, $\Gamma(x)$ is integrable. This in particular implies that all integrals in Eq. (4.31) exist. Note that in the $Q^2 \rightarrow \infty$ limit the Δ prefactor vanishes, consistent with asymptotic freedom.

B. Statistical estimators

We define statistical estimators that are used to assess the quality of both the Monte Carlo replica generation (replica averages) and the neural network training (neural network averages and statistical distance). The superscripts (dat), (art) and (net) refer respectively to the original data, to the N_{rep} Monte Carlo replicas of the data, and to the N_{rep} neural networks. The subscripts rep and dat refer respectively to whether averages are taken by summing over all replicas or over all data.

• Replica averages

- Average over the number of replicas for each experimental point i

$$\left\langle F_i^{(\text{art})} \right\rangle_{\text{rep}} = \frac{1}{N_{\text{rep}}} \sum_{k=1}^{N_{\text{rep}}} F_i^{(\text{art})(k)} . \quad (\text{B.1})$$

- Associated variance

$$\sigma_i^{(\text{art})} = \sqrt{\frac{N_{\text{rep}}}{N_{\text{rep}} - 1} \left(\left\langle \left(F_i^{(\text{art})} \right)^2 \right\rangle_{\text{rep}} - \left\langle F_i^{(\text{art})} \right\rangle_{\text{rep}}^2 \right)} . \quad (\text{B.2})$$

- Associated covariance

$$\rho_{ij}^{(\text{art})} = \frac{N_{\text{rep}}}{N_{\text{rep}} - 1} \frac{\langle F_i^{(\text{art})} F_j^{(\text{art})} \rangle_{\text{rep}} - \langle F_i^{(\text{art})} \rangle_{\text{rep}} \langle F_j^{(\text{art})} \rangle_{\text{rep}}}{\sigma_i^{(\text{art})} \sigma_j^{(\text{art})}} . \quad (\text{B.3})$$

$$\text{cov}_{ij}^{(\text{art})} = \rho_{ij}^{(\text{art})} \sigma_i^{(\text{art})} \sigma_j^{(\text{art})} . \quad (\text{B.4})$$

- Percentage error on central values over the N_{dat} data points.

$$\langle PE \left[\langle F^{(\text{art})} \rangle_{\text{rep}} \right] \rangle_{\text{dat}} = \frac{1}{N_{\text{dat}}} \sum_{i=1}^{N_{\text{dat}}} \left[\frac{\langle F_i^{(\text{art})} \rangle_{\text{rep}} - F_i^{(\text{exp})}}{F_i^{(\text{exp})}} \right] . \quad (\text{B.5})$$

We define analogously $\langle PE \left[\langle \sigma^{(\text{art})} \rangle_{\text{rep}} \right] \rangle_{\text{dat}}$.

- Scatter correlation:

$$r \left[F^{(\text{art})} \right] = \frac{\langle F^{(\text{exp})} \langle F^{(\text{art})} \rangle_{\text{rep}} \rangle_{\text{dat}} - \langle F^{(\text{exp})} \rangle_{\text{dat}} \langle \langle F^{(\text{art})} \rangle_{\text{rep}} \rangle_{\text{dat}}}{\sigma_s^{(\text{exp})} \sigma_s^{(\text{art})}} \quad (\text{B.6})$$

where the scatter variances are defined as

$$\sigma_s^{(\text{exp})} = \sqrt{\langle (F^{(\text{exp})})^2 \rangle_{\text{dat}} - (\langle F^{(\text{exp})} \rangle_{\text{dat}})^2} \quad (\text{B.7})$$

$$\sigma_s^{(\text{art})} = \sqrt{\langle (\langle F^{(\text{art})} \rangle_{\text{rep}})^2 \rangle_{\text{dat}} - (\langle \langle F^{(\text{art})} \rangle_{\text{rep}} \rangle_{\text{dat}})^2} . \quad (\text{B.8})$$

We define analogously $r \left[\sigma^{(\text{art})} \right]$, $r \left[\rho^{(\text{art})} \right]$ and $r \left[\text{cov}^{(\text{art})} \right]$. Note that the scatter correlation and scatter variance are not related to the variance and correlation Eqs. (B.2)-(B.4).

- Average variance:

$$\langle \sigma^{(\text{art})} \rangle_{\text{dat}} = \frac{1}{N_{\text{dat}}} \sum_{i=1}^{N_{\text{dat}}} \sigma_i^{(\text{art})} . \quad (\text{B.9})$$

We define analogously $\langle \rho^{(\text{art})} \rangle_{\text{dat}}$ and $\langle \text{cov}^{(\text{art})} \rangle_{\text{dat}}$, as well as the corresponding experimental quantities.

• Neural network averages

- Average error over networks

$$\langle E \rangle = \frac{1}{N_{\text{rep}}} \sum_{k=1}^{N_{\text{rep}}} E^{(k)} , \quad (\text{B.10})$$

where $E^{(k)}$ is given by Eq. (5.5).

- Average over neural networks for each experimental point i

$$\langle F_i^{(\text{net})} \rangle_{\text{rep}} = \frac{1}{N_{\text{rep}}} \sum_{k=1}^{N_{\text{rep}}} F_i^{(\text{net})(k)}, \quad (\text{B.11})$$

where $F_i^{(\text{net})}$ is the value of the nonsinglet structure function computed using Eq. (4.11) from the neural network parametrization of the nonsinglet quark distribution at the values of x and Q^2 which correspond to the i -th data point

- Associated variance

$$\sigma_i^{(\text{net})} = \sqrt{\frac{N_{\text{rep}}}{N_{\text{rep}} - 1} \left(\langle (F_i^{(\text{net})})^2 \rangle_{\text{rep}} - \langle F_i^{(\text{net})} \rangle_{\text{rep}}^2 \right)}. \quad (\text{B.12})$$

- Associated covariance

$$\rho_{ij}^{(\text{net})} = \frac{N_{\text{rep}}}{N_{\text{rep}} - 1} \frac{\langle F_i^{(\text{net})} F_j^{(\text{net})} \rangle_{\text{rep}} - \langle F_i^{(\text{net})} \rangle_{\text{rep}} \langle F_j^{(\text{net})} \rangle_{\text{rep}}}{\sigma_i^{(\text{net})} \sigma_j^{(\text{net})}}. \quad (\text{B.13})$$

$$\text{cov}_{ij}^{(\text{net})} = \rho_{ij}^{(\text{net})} \sigma_i^{(\text{net})} \sigma_j^{(\text{net})}. \quad (\text{B.14})$$

The corresponding means over the N_{dat} data points are computed using the analogue of Eq. (B.9).

- Scatter correlation

$$r [F^{(\text{net})}] = \frac{\langle F^{(\text{exp})} \langle F^{(\text{net})} \rangle_{\text{rep}} \rangle_{\text{dat}} - \langle F^{(\text{exp})} \rangle_{\text{dat}} \langle \langle F^{(\text{net})} \rangle_{\text{rep}} \rangle_{\text{dat}}}{\sigma_s^{(\text{exp})} \sigma_s^{(\text{net})}}. \quad (\text{B.15})$$

We define analogously $\langle \rho^{(\text{net})} \rangle_{\text{dat}}$ and $\langle \text{cov}^{(\text{net})} \rangle_{\text{dat}}$.

• Distance between samples

- Distance between different sets of neural nets

$$\langle d[q] \rangle = \sqrt{\left\langle \frac{\left(\langle q_i \rangle_{(1)} - \langle q_i \rangle_{(2)} \right)^2}{\sigma^2[q_i^{(1)}] + \sigma^2[q_i^{(2)}]} \right\rangle_{\text{dat}}}, \quad (\text{B.16})$$

with

$$\langle q_i \rangle_{(1)} = \frac{1}{N_{\text{rep}}^{(1)}} \sum_{k=1}^{N_{\text{rep}}^{(1)}} q_{ik}^{(1)} \quad (\text{B.17})$$

$$\sigma^2[q_i^{(1)}] = \frac{1}{N_{\text{rep}}^{(1)}(N_{\text{rep}}^{(1)} - 1)} \sum_{k=1}^{N_{\text{rep}}^{(1)}} \left(q_{ik}^{(1)} - \langle q_i \rangle_{(1)} \right)^2, \quad (\text{B.18})$$

where $q_{ik}^{(1)}$ is the value of the nonsinglet quark distribution for the i -th data point computed for the k -th replica from a set of $N_{\text{rep}}^{(1)}$ neural networks, and likewise $q_{ik}^{(2)}$ is the value computed for the k -th replica from a second set of $N_{\text{rep}}^{(2)}$ neural networks.

- The distance between errors and correlations is defined using Eq. (B.16), but with the average value Eq. (B.17) replaced by the variance

$$\sigma_i^{(\text{net})} = \sqrt{\frac{N_{(1)}}{N_{(1)} - 1} \left(\langle (q_i)^2 \rangle_{(1)} - \langle q_i \rangle_{(1)}^2 \right)}, \quad (\text{B.19})$$

and similarly for the covariance. Note that the distance is just the square root of the sample variance normalized to the variance of which it is an unbiased estimator (see Ref. [15], in particular for a discussion of the variance of d).

- **Pull**

If $F_2^{(a)}(x, Q^2)$ and $F_2^{(b)}(x, Q^2)$ are two different determinations of the structure function, each affected respectively by uncertainty $\sigma_{(a)}(x, Q^2)$ and $\sigma_{(b)}(x, Q^2)$, the pull is defined as

$$\langle P(x, Q^2) \rangle = \frac{\left(F_2^{(a)}(x, Q^2) - F_2^{(b)}(x, Q^2) \right)}{\sqrt{\sigma_{(a)}^2(x, Q^2) + \sigma_{(b)}^2(x, Q^2)}}. \quad (\text{B.20})$$

References

- [1] S. Catani *et al.*, *Qcd*, [hep-ph/0005025](#).
- [2] M. Dittmar *et al.*, *Parton distributions: Summary report*, [hep-ph/0511119](#).
- [3] S. Forte, *Structure functions and parton distributions*, *Nucl. Phys.* **A755** (2005) 100–110, [[hep-ph/0502073](#)].
- [4] S. Alekhin, *Parton distributions from deep-inelastic scattering data*, *Phys. Rev.* **D68** (2003) 014002, [[hep-ph/0211096](#)].
- [5] J. Pumplin *et al.*, *New generation of parton distributions with uncertainties from global qcd analysis*, *JHEP* **07** (2002) 012, [[hep-ph/0201195](#)].
- [6] A. D. Martin, R. G. Roberts, W. J. Stirling, and R. S. Thorne, *Uncertainties of predictions from parton distributions. i: Experimental errors. ((t))*, *Eur. Phys. J.* **C28** (2003) 455–473, [[hep-ph/0211080](#)].
- [7] W. T. Giele, S. A. Keller, and D. A. Kosower, *Parton distribution function uncertainties*, [hep-ph/0104052](#).
- [8] A. Djouadi and S. Ferrag, *Pdf uncertainties in higgs production at hadron colliders*, *Phys. Lett.* **B586** (2004) 345–352, [[hep-ph/0310209](#)].
- [9] S. Forte, L. Garrido, J. I. Latorre, and A. Piccione, *Neural network parametrization of deep-inelastic structure functions*, *JHEP* **05** (2002) 062, [[hep-ph/0204232](#)].
- [10] **NNPDF** Collaboration, L. Del Debbio, S. Forte, J. I. Latorre, A. Piccione, and J. Rojo, *Unbiased determination of the proton structure function $f_2(p)$ with faithful uncertainty estimation*, *JHEP* **03** (2005) 080, [[hep-ph/0501067](#)].
- [11] J. Rojo and J. I. Latorre, *Neural network parametrization of spectral functions from hadronic tau decays and determination of qcd vacuum condensates*, *JHEP* **01** (2004) 055, [[hep-ph/0401047](#)].

- [12] J. Rojo, *Neural network parametrization of the lepton energy spectrum in semileptonic b meson decays*, *JHEP* **05** (2006) 040, [[hep-ph/0601229](#)].
- [13] M. C. Gonzalez-Garcia, M. Maltoni, and J. Rojo, *Determination of the atmospheric neutrino fluxes from atmospheric neutrino data*, *JHEP* **10** (2006) 075, [[hep-ph/0607324](#)].
- [14] G. D’Agostini, *Bayesian reasoning in data analysis: A critical introduction*, . New Jersey, USA: World Scientific (2003) 329 p.
- [15] G. Cowan, *Statistical data analysis*, *Oxford Science Publications* (2002).
- [16] M. Diemoz, F. Ferroni, E. Longo, and G. Martinelli, *Parton densities from deep inelastic scattering to hadronic processes at super collider energies*, *Z. Phys.* **C39** (1988) 21.
- [17] **E665** Collaboration, M. R. Adams *et al.*, *Proton and deuteron structure functions in muon scattering at 470-gev*, *Phys. Rev.* **D54** (1996) 3006–3056.
- [18] **New Muon** Collaboration, M. Arneodo *et al.*, *Measurement of the proton and deuteron structure functions, $f_2(p)$ and $f_2(d)$, and of the ratio $\sigma(l)/\sigma(t)$* , *Nucl. Phys.* **B483** (1997) 3–43, [[hep-ph/9610231](#)].
- [19] **BCDMS** Collaboration, A. C. Benvenuti *et al.*, *A high statistics measurement of the proton structure functions $f_2(x, q^{*2})$ and r from deep inelastic muon scattering at high q^{*2}* , *Phys. Lett.* **B223** (1989) 485.
- [20] **BCDMS** Collaboration, A. C. Benvenuti *et al.*, *A high statistics measurement of the deuteron structure functions $f_2(x, q^{*2})$ and r from deep inelastic muon scattering at high q^{*2}* , *Phys. Lett.* **B237** (1990) 592.
- [21] J. A. M. Vermaseren, A. Vogt, and S. Moch, *The third-order qcd corrections to deep-inelastic scattering by photon exchange*, *Nucl. Phys.* **B724** (2005) 3–182, [[hep-ph/0504242](#)].
- [22] S. Moch, J. A. M. Vermaseren, and A. Vogt, *The three-loop splitting functions in qcd: The non-singlet case*, *Nucl. Phys.* **B688** (2004) 101–134, [[hep-ph/0403192](#)].
- [23] A. Vogt, *Efficient evolution of unpolarized and polarized parton distributions with qcd-pegasus*, *Comput. Phys. Commun.* **170** (2005) 65–92, [[hep-ph/0408244](#)].
- [24] M. Botje, *Qcdnum manual*, <http://www.nikhef.nl/~h24/qcdnum/>.
- [25] G. P. Salam, *Hoppet manual*, <http://hepforge.cedar.ac.uk/hoppet/>.
- [26] **Particle Data Group** Collaboration, W. M. Yao *et al.*, *Review of particle physics*, *J. Phys.* **G33** (2006) 1–1232.
- [27] K. G. Chetyrkin, B. A. Kniehl, and M. Steinhauser, *Strong coupling constant with flavour thresholds at four loops in the \overline{ms} scheme*, *Phys. Rev. Lett.* **79** (1997) 2184–2187, [[hep-ph/9706430](#)].
- [28] R. S. Thorne, *A variable-flavour number scheme for nnlo*, *Phys. Rev.* **D73** (2006) 054019, [[hep-ph/0601245](#)].
- [29] J. Abate and P. Valko, *Multi-precision laplace transform inversion*, *International Journal for Numerical Methods in Engineering* **60** (2003) 979–993.
- [30] W. Giele *et al.*, *The qcd/sm working group: Summary report*, [hep-ph/0204316](#).
- [31] H. Georgi and H. D. Politzer, *Freedom at moderate energies: Masses in color dynamics*, *Phys. Rev.* **D14** (1976) 1829.

- [32] R. G. Roberts, *The Structure of the proton: Deep inelastic scattering*. Cambridge, UK: Univ. Pr., 1990.
- [33] Y. A. Kanev, *Application of neural networks and genetic algorithms in high-energy physics*, . UMI-99-05968.
- [34] L. Lonnblad, C. Peterson, and T. Rognvaldsson, *Using neural networks to identify jets*, *Nucl. Phys.* **B349** (1991) 675–702.
- [35] M. Gluck, E. Reya, and C. Schuck, *Non-singlet qcd analysis of $f_2(x, q^{*2})$ up to nnlo*, *Nucl. Phys.* **B754** (2006) 178–186, [hep-ph/0604116].
- [36] J. Blumlein, H. Bottcher, and A. Guffanti, *Non-singlet qcd analysis of deep inelastic world data at $o(\alpha(s)^{*3})$* , hep-ph/0607200.
- [37] S. Forte and G. Ridolfi, *Renormalization group approach to soft gluon resummation*, *Nucl. Phys.* **B650** (2003) 229–270, [hep-ph/0209154].