Eur. Phys. J. C (2022) 82:163
https://doi.org/10.1140/epjc/s10052-022-10136-z

THE EUROPEAN
PHYSICAL JOURNAL C

Regular Article - Theoretical Physics

A data-based parametrization of parton distribution functions

Stefano Carrazza[1,2,3,a], **Juan Cruz-Martinez**[1], **Roy Stegeman**[1]

[1] TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano and INFN Sezione di Milano, Milan, Italy
[2] Theoretical Physics Department, CERN, 1211 Geneva 23, Switzerland
[3] Quantum Research Centre, Technology Innovation Institute, Abu Dhabi, United Arab Emirates

Received: 5 November 2021 / Accepted: 11 February 2022

Abstract Since the first determination of a structure function many decades ago, all methodologies used to determine structure functions or parton distribution functions (PDFs) have employed a common prefactor as part of the parametrization. The NNPDF collaboration pioneered the use of neural networks to overcome the inherent bias of constraining the space of solution with a fixed functional form while still keeping the same common prefactor as a preprocessing. Over the years various, increasingly sophisticated, techniques have been introduced to counter the effect of the prefactor on the PDF determination. In this paper we present a methodology to perform a data-based scaling of the Bjorken x input parameter which facilitates the removal the prefactor, thereby significantly simplifying the methodology, without a loss of efficiency and finding good agreement with previous results.

1 Introduction

Parton distribution functions (PDFs) provide a description of the non-perturbative structure of hadrons [1–3]. An accurate and precise description of PDFs is needed to make theoretical predictions for precision physics at hadron colliders. Since PDFs are non-perturbative, only lattice QCD allows for a numerical approach to the problem [4], otherwise it is not possible to obtain the PDFs directly from first principles. As a consequence, they are determined by performing fits to experimental data. This requires the use of a functional form of which parameters are to be fitted. Such a procedure can potentially lead to a biased result, in some cases ultimately resulting in underestimated PDF uncertainties [5].

PDF fitting collaborations have developed different techniques to gauge and reduce the impact of the different sources of bias and provide a correct estimate of PDF uncertainties. Collaborations using the Hessian approach introduce the notion of "tolerance", whereby the uncertainties are rescaled by a global factor to obtain better agreement with the experimental data. It should be noted however, that tolerance is used to account for all sources of uncertainty, not just that corresponding to the parametrization but in particular it is also used to account for tensions between datasets and theoretical uncertainties. On the other hand, the NNPDF collaboration addresses the problem of a biased functional form by parametrizing the PDFs with neural networks. Neural networks can represent any function, as is understood through the universal approximation theorems [6], thus avoiding biasing the PDF trough the choice of the functional form.

In practical terms, however, the training of the parameters of a neural network occurs in a finite number of steps and some assumptions are to be introduced to improve the efficiency of the training. In the specific case of NNPDF, the input values are scaled according to their expected distribution and the neural network is multiplied by a prefactor designed to speed up the training. Great care has been put into ensuring that such preprocessing does not introduce a bias in the PDF determination [7] but their effect needs to be reevaluated with every change to the methodology or dataset [5,8,9].

In this paper we present a purely data-driven approach for the preprocessing of the input data. We implement it as part of the latest NNPDF fitting framework [10] and analyze the resulting PDFs. In both the data region and the extrapolation regions we obtain results compatible with those of NNPDF4.0 [11], with similar rates of convergence. This preprocessing methodology brings two important advantages, firstly it facilitates the use of the NNPDF open source framework [12] by external users by automatizing steps that until now required human intervention and, secondly, it validates the NNPDF4.0 determination by removing two possible sources of bias or inefficiencies without a significant change of the results.

[a] e-mail: stefano.carrazza@cern.ch (corresponding author)

Published online: 22 February 2022

The first possible source of bias is due to the input layer: in all previous releases of NNPDF (see e.g. Refs. [9,11,13] for the most recent releases) the input layer takes the momentum fraction x and splits it into a tuple $(x, \log x)$ which is provided as input to the dense layers of the neural network. As will be discussed in more detail in Sect. 3 the non-trivial interplay between the stopping algorithm and the rates of convergence of two different input scales can lead to overfitted or underfitted scenarios, putting an extra burden on the hyperoptimization algorithm.

The second potential source of bias is the prefactor $x^{(1-\alpha_a)}(1-x)^{\beta_a}$ used by all PDF fitting groups as part of their parametrization of the PDFs. For most groups, such as MSHT [14], CTEQ [15], and ABMP [16], this prefactor is used as part of the functional form of the fitted PDFs (and thus its parameters are also fitted). In the case of NNPDF instead the exponential parameters, α_a and β_a, are chosen randomly in a per-replica basis from a pre-established range [8,13]. The random selection of exponents was introduced due to the observation in Ref. [5] that uncertainties could be underestimated if the exponential parameters were to be fixed. While as a result of this mitigation strategy the prefactor becomes just a tool to speed-up convergence, it has a negative impact on aspects of the methodology such as the scan of hyperparameters. In Sect. 4 we will discuss how, as a consequence of the improvements of NNPDF4.0, the prefactor can be removed from the methodology and thus also the mitigation strategies related to it.

The methodology we present is a feature scaling of the input x, as is standard practice in the machine learning community, leading to a simplification of the neural network architecture. We show that PDFs resulting from these changes are faithful both in the data and the extrapolation regions, and that they are compatible with the current generation of NNPDF fits.

The paper is structured as follows. In Sect. 2 we will highlight in more detail the parametrization choices made by PDF fitting collaborations. In this section we will in particular focus on the NNPDF methodology, as the main purpose of this paper is to present the feasibility of the approach by applying it to the NNPDF framework. Then, in Sect. 3 we will discuss the new input scaling, and in Sect. 4 we will discuss how this change to the input scaling allows for the removal of the prefactor. Finally in Sect. 5 we validate the accuracy and faithfulness of the PDFs by performing various tests which fully validate the presented methodology.

2 Parametrization of PDFs

All of the most used PDF sets are parametrized at some input scale Q_0 by a function of the form

$$xf_a(x, Q_0) = A_a x^{(1-\alpha_a)}(1-x)^{\beta_a}\mathcal{P}_a(y(x)), \tag{1}$$

where the indices a correspond to the type of parton, and $\mathcal{P}_a$ is a functional form that is different between PDF fitting groups, with an input $y(x)$ that differs between PDF sets. In particular, for the most recent PDF sets released by the MSHT [14], CTEQ [15], and ABMP [16] collaborations $\mathcal{P}_a$ represents a polynomial per flavor, while for the latest PDF set released by the NNPDF collaboration [11] $\mathcal{P}_a$ is represented by a single neural network with one output node per flavor.

The PDFs are kinematically constrained by

$$f_a(x=1, Q) = 0, \tag{2}$$

which is enforced through the $(1-x)^{\beta_a}$ component in (1). The motivation to choose this functional form stems from the constituent counting rules [17]. In the fitting methodologies this component not only ensures that the condition of (2) is satisfied, but it partially controls the large-x extrapolation region where data is unavailable. The small-x behavior instead is controlled by the prefactor $x^{(1-\alpha_a)}$. The introduction of this factor was inspired by Regge theory [18]. While enforcing this behavior implies a methodological bias [19], studies on the extrapolation behavior of PDF determinations have found a qualitative agreement with the expected values [20].

Despite its generalized use, a fixed functional form as prefactor introduces two issues in the determination of PDFs. First, the PDFs are only based on data in the domain $10^{-5} \lesssim x \lesssim 0.75$ while PDF grids are delivered in the domain $10^{-9} \leq x \leq 1$ and as a result there is a potential bias in the extrapolation regions. Secondly, even if we assume that outside the data region the PDFs are described by the given functional form, it is not clear at which scale Q^2 it should hold and it is not preserved under Q^2 transformation.

In this paper we will focus on the implications of this parametrization in the context of the NNPDF methodology, thus where the PDFs are parametrized with a neural network of which the output nodes correspond to a linear combination of PDF flavors at an energy scale Q_0. The neural network consists of fully-connected layers where the depth of the network and the number of nodes in each layer, as well as the type of activation function, are determined through a hyperoptimization procedure [10].

The NNPDF model then can be written as

$$xf_a(x, Q_0) = A_a x^{(1-\alpha_a)}(1-x)^{\beta_a}\mathrm{NN}_a(x), \tag{3}$$

where we recognize (1) with the general function $\mathcal{P}_a$ replaced by a neural network NN_a. In the standard PDF fit the output flavors a correspond the so-called evolution basis, a linear combinations of PDF flavors that are eigenstates of the Q^2 evolution [11], however, in Sect. 8.4 of Ref. [11] it has been shown that different choices of the parametrization basis give good agreement within PDF uncertainties. The prefactor A_a

is a normalization constant to ensure that the momentum and valence sum rules are satisfied.

The effect of the exponents α_a and β_a as a source of bias [5] is mitigated in the NNPDF methodology by randomly sampling them from a uniform distribution, the boundaries of which are determined through an iterative procedure [8,13].

The diagram in Fig. 1 shows how, within the NNPDF fitting framework, the figure of merit χ^2 is evaluated for an input grid $\{x_n^p\}$, and how the prefactor and neural network are combined to produce unnormalized PDFs $\tilde{f}_a(x_n^p)$, which are then normalized by enforcing the momentum and valence sum rules [11]. The output of the parametrization then corresponds to normalized PDFs at an input scale Q_0.

The first layer of the network maps the input node (x) onto the pair $(x, \log x)$. The choice for this splitting of the input results from the observation that typically PDFs show logarithmic behavior at small-x ($x \lesssim 0.01$) and linear behavior at large-x ($x \gtrsim 0.01$) [7], and together with the prefactor it ensures convergence of the optimization algorithm in the small-x region.

The PDFs themselves do not allow for a direct comparison to data, instead we convolute the PDFs with partonic cross-sections to calculate a theoretical prediction of physical observables corresponding to the experimental measurements. These partonic cross-sections, along with the QCD evolution equations, are encoded in `FastKernel` (FK) tables [21,22], allowing for an efficient computation of the relevant observables. For hadronic observables the corresponding calculation is

$$\mathcal{O}_n = \mathrm{FK}^n_{abpq} f_a(x_n^p, Q_0) f_b(x_n^q, Q_0), \tag{4}$$

while for deep inelastic scattering observables it reduces to

$$\mathcal{O}_n = \mathrm{FK}^n_{ap} f_a(x_n^p, Q_0), \tag{5}$$

where n labels the experimental datapoints, a and b the PDF flavors, and p and q the point in the corresponding x-grid.

After calculating the observables as predicted by the PDFs, the figure of merit minimized during a fit is defined as

$$\chi^2 = \sum_{i,j}^{N_{\mathrm{dat}}} (D-P)_i C_{ij}^{-1} (D-P)_j, \tag{6}$$

with

$$C_{ij} = C_{ij}^{(\mathrm{exp})}. \tag{7}$$

Here D_i is the experimentally determined value of the i-th datapoint, P_i is the theoretical prediction of the i-th datapoint calculated using (4) or (5), and $C_{ij}^{(\mathrm{exp})}$ is the covariance matrix of the experimental dataset. For the optimization of the neural network the covariance matrix is treated following the t_0 prescription of Ref. [23].

The final predictions (and experimental data) are then split into training and validation sets such that only the χ^2 of the trained data is used for the minimization while the validation χ^2 is utilized for early stopping.

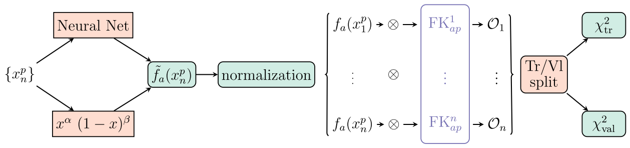

Fig. 1 Diagrammatic representation of the calculation of the χ^2 in the NNPDF fitting framework as a function of the values of $\{x_n^p\}$ for the different datasets. Each block indicates an independent component

3 A data-based scaling of the input x-grids

Often, in machine learning problems, the input data can be unbalanced or span several orders of magnitude. Such is the case of PDF fitting, where the input is concentrated at small-x.

This can be a problem because, as we will explicitly show below, having input features of different magnitudes introduces an artificial impact on the importance of each feature within the network. This problem is exacerbated for gradient descent based algorithms where the issue propagates to the learning rate of the weights of the network. Thus, even if the algorithm is still able to find the global minimum, the rate of convergence is not equal for all features. In the case we are interested in (the NNPDF methodology with an early stopping algorithm) this can lead to locally overfitted or underfitted results in different regions of the kinematic domain. Ideally the fitting methodology should result in a uniform rate of convergence across all input scales.

In short, the problem is that while the data spans multiple orders of magnitude, the fitting methodology requires the inputs to be of the same length scale. Below we discuss the impact of the input scaling on the PDFs, and provide a methodology that takes an arbitrary input grid and scales it such that the optimizer always has a good resolution across the entire input grid.

At this point one may note that the input x-grids to the neural network are the grids defined in the FK-tables as shown in Eqs. (4), (5), which may differ from the x-values of the corresponding experimental datasets. While this is true, from the perspective of the fitting methodology, the grid choice is arbitrary and thus the problem remains.

In NNPDF fits, the input variable is mapped to $(x, \log x)$ in the first layer of the neural network which facilitates the methodology in learning features of the PDF that scale either linearly or logarithmically in x. This splitting was first introduced in Ref. [7] and was motivated by the expectation that they are the variables upon which the structure functions F_2 depend. It was noted that, by merit of the neural network-

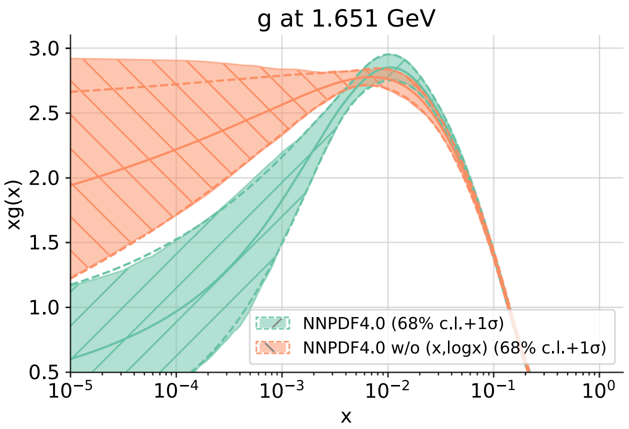

Fig. 2 Comparison between the gluon PDF generated with the standard NNPDF4.0 methodology (green) and our modification in which we have removed the splitting layer of x to $(x, \log x)$ (orange). While we observe good compatibility between both PDFs in the large-x region, as we enter in the small-x region our modified PDF saturates. This is evident also in the χ^2 of the modified fit which was blocked at $\chi^2 = 1.20$ while NNPDF4.0 is able to get it down to $\chi^2 = 1.16$

based parametrization, the choice of input scales could affect the rate of convergence but not the final result.

However, it can be seen in Fig. 2 that the $(x, \log x)$ split can have an effect on the shape of the PDFs. In Fig. 2 we compare the gluon PDF of the NNPDF4.0 fit to a PDF generated using the same data, theory and methodological settings, but with the $(x, \log x)$ input scaling replaced with only an (x) input.[1]

While the NNPDF4.0 methodology was sensitive to the small-x region (where the logarithmic behavior is expected) when we remove $(\log x)$ from the input we can observe a hint of saturation in said region. Despite the fact that the (x) and $(\log x)$ variables contain the same information the split has a noticeable effect on the fit, in this case speeding up the minimization process in the small-x region.

In order for the optimization algorithm to be able to easily learn features across many orders of magnitude we can perform a feature scaling of the training input x such that the distances between all points are of the same order of magnitude. In particular, we can consider mapping the combined training input x-grid from the FK-tables of all datasets as discussed in Sect. 2 to an empirical cumulative distribution function (eCDF) of itself. The eCDF is defined as a step function that starts at 0 and increases by $1/N_x$ at each point of the input x-grids, with N_x the total number of nodes in the x-grids. If the x-grids of n FK-tables share a common point in x, the step-size corresponding to this point is instead n/N_x. This results in a function whose value at any x corresponds to the fraction of points in the x-grids that are less than or equal to x. In other words, while the x values present in the FK table are not uniformly distributed on the domain $0 \leq x \leq 1$, applying the eCDF makes it that they are. A density plot of the distribution of input points without scaling, logarithmically scaled, and after applying the eCDF is shown in Fig. 3. This figure also clearly shows that both inputs to the neural network as used in NNPDF4.0 [11] have a high density of points on the same scale.

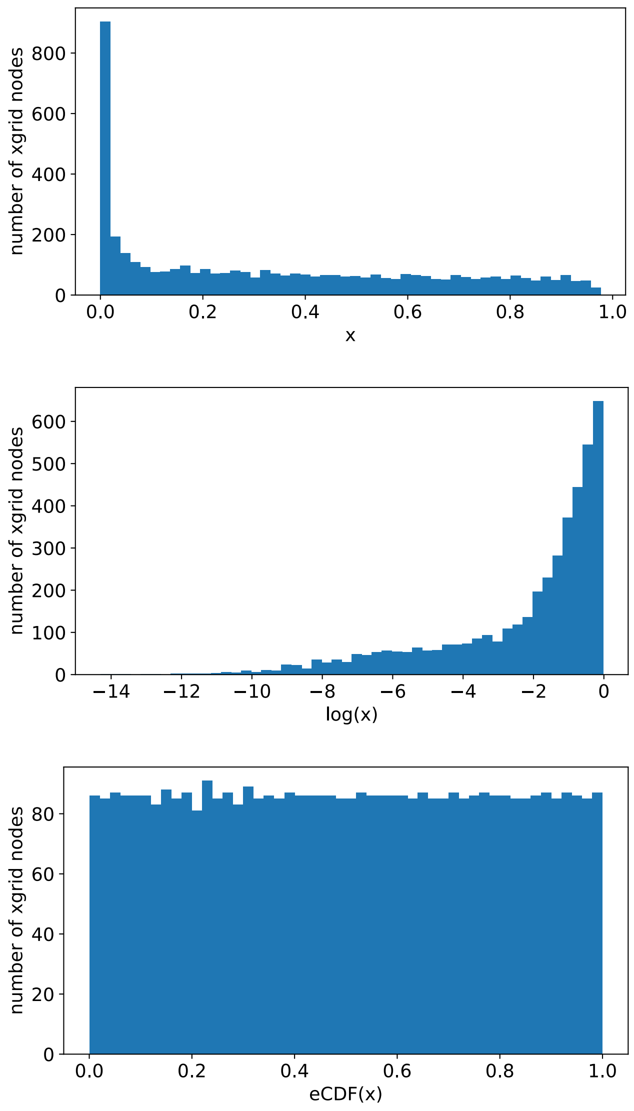

Fig. 3 Histograms showing the distribution of the unscaled x points in the FK-table x-grids (top), as well as the distribution of the input points after scaling with $\log x$ (middle) and eCDF (bottom)

Applying the eCDF results in a distribution on the domain $0 \leq x \leq 1$. However, for the results presented in this paper the eCDF transformation is followed by a linear scaling, resulting in a total transformation of the input $\hat{x} = 2 \cdot \text{eCDF(x)} - 1$, meaning that the input values to the neural network are in the range $-1 \leq x \leq 1$. This is done to ensure that the input is symmetric around 0 which results

[1] This plot, as well as the other PDF plots in this paper, have been produced with the `validphys` package [12,24].

in improved convergence for many of the commonly used activation functions in neural networks.

Since using the eCDF means that we apply a discrete scaling only for values present in the input x-grids, we need to also add both an interpolation and an extrapolation function to extract PDF values at values of the momentum fraction that do not coincide with the input x-grids. Here it is important to note that the PDFs are made publicly available through the LHAPDF interface, and that they are correspondingly stored in the LHAPDF grid format [25]. Because LHAPDF grids are provided on the domain $10^{-9} \leq x \leq 1$, the problem of extrapolation can be turned into an interpolation problem by including the points $x = 10^{-9}$ and $x = 1$ in the input x-grid before determining the eCDF, and defining a methodology for interpolation.

The simplest option for an interpolation function is a "nearest neighbor" mapping, whereby we map any input on the continuous domain $0 \leq x \leq 1$ to the nearest node in the x-grids of the FK-table.

We can a nevertheless improve this simple mapping by instead using a continuous function. A requirement of any such interpolation function is that it needs to be monotonically increasing. However, if we determine the interpolation between each two points of the FK-table x-grids the optimization algorithm will be agnostic to the existence of this interpolation function as it is never probed. Ideally, in particular for the evaluation of validation data of which the corresponding FK-tables were not included when defining the eCDF scaling, we want the optimizer to probe the interpolation functions such that it is able to learn its properties and as a result provide a more accurate prediction in the interpolation region as well.

As such, the interpolation functions are not defined between each neighboring pair of values in the input x-grid, but rather we select N_{int} evenly distributed points (after the eCDF transformation) between which to define interpolation functions. Here N_{int} is a new hyperparameter, though not necessarily one that needs to be free during hyperoptimization of the methodology. To obtain a monotonic interpolation function, we propose determining the interpolation functions using cubic Hermite splines [26].

By scaling the input as discussed in this section we remove any restrictions on the PDF resulting from the input features, while simultaneously simplifying the model architecture by getting rid of the mixing of two different orders of magnitude in the first layer. In Fig. 4 we compare the gluon PDF generated using the NNPDF4.0 methodology, to a PDF generated using the same data and theory settings, but with the $(x, \log x)$ input scaling replaced with the eCDF input scaling as described above. This comparison of the gluon PDF is representative for all flavors, and shows that the PDFs produced with this new scaling are in agreement with those found using the $(x, \log x)$ input. If the PDFs had not been in agreement that would have suggested that the PDFs had a component that scaled neither linearly nor logarithmically which the neural network was not able to accommodate.

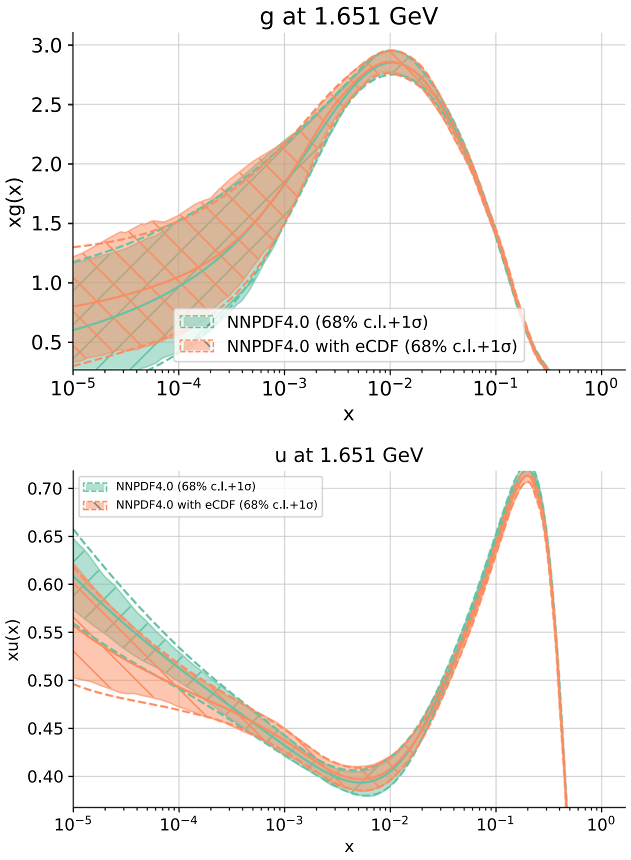

Fig. 4 Comparison between the gluon and up PDFs determined using the NNPDF4.0 methodology (green) and a PDF determined using input scaling based on the eCDF (orange) with all other parameters the same

4 Removing the prefactor

In Sect. 3 we discuss a new way of treating the input for the PDF fitting by rescaling the input in a systematic way that depends only on the fitted data itself. This is a purely data-driven approach and thus free of sources of bias due to the choice of functional form. As we will show here, the scaling of the input grid in x by using the eCDF will also allow us to remove the prefactor entirely.

In what follows we will discuss the consequence of removing the prefactor. Specifically, by "removing the prefactor", we understand a treatment which is equivalent to setting $\alpha_a = 1$ and $\beta_a = 0$ in (1), while enforcing the condition of (2). As a result the PDF model is written as

$$x f_a(x, Q_0) = A_a \left[\mathrm{NN}_a(x) - \mathrm{NN}_a(1)\right]. \tag{8}$$

A similar model, without the model-agnostic input scaling, has previously been applied to the study of fragmentation functions [27]. We will focus on the effects of the change in the small-x and large-x extrapolation regions where the lack of data makes the fit particularly prone to methodological biases.

In Sect. 2 it was mentioned that the motivation to include the prefactor in NNPDF is to improve convergence during optimization and that its effect as a source of bias in the extrapolation region was mitigated by randomly sampling the exponents α_a and β_b. However, while this makes the fit robust with respect to the exponents it still comes with a cost: it introduces an additional source of fluctuations between replicas which can be undesirable in certain cases and limit progress around the methodological development.

As a result of switching to optimizers based on stochastic gradient descent (SGD) for NNPDF4.0, instead of using the genetic algorithm used for NNPDF3.1, the average time to fit a replica has been reduced by an order of magnitude, and stability has greatly improved [10,11]. These improvements of the optimization algorithm allow us to remove the prefactor without a significant change in computational costs and therefore any possible benefit of the prefactor in terms of convergence no longer outweighs its disadvantages.

As an example of where fluctuations between replicas as a result of the randomized exponents of the prefactor can limit the development of the methodology, one can consider the hyperoptimization procedure introduced in Ref. [10] and further developed in Ref. [11]. The hyperoptimization algorithm employs out-of-sample testing through K-folds cross-validation. In the current scenario an otherwise good hyperparameter setup with poor exponents in the prefactor can return a worse figure of merit during hyperparameter optimization than a relatively poorer hyperparameter setup with very suitable exponents. As a result many more hyperparameter combinations need to be tested to overcome the statistical noise. Removing the replica-by-replica random sampling of the exponents removes this effect from hyperoptimization.

The uncertainties of the fit in the extrapolation region are closely related to the ranges the prefactor exponents are sampled from. Removing them from the parametrization also removes the random sampling, it is therefore important to validate the obtained small-x and large-x uncertainties as will be done in Sect. 5.

For brevity and clarity, we will from now on refer to the proposed methodology without the prefactor and with the eCDF input scaling as the "feature scaling" methodology.

5 Results and validation

5.1 Tuning the methodology: hyperoptimization

After any significant change to the fitting methodology, it is important to re-evaluate the choice of the hyperparameters of the model. The model parameters obtained through the hyperoptimization procedure are given in Table 1. The details of the procedure are described in Sect. 3.3 of Ref. [11]. Note that the selected activation function does not saturate asymptotically for large or small values of x, preventing saturation in the extrapolation region. The choice of activation function was however not fixed during the selection of hyperparameters, this activation function has been selected by the hyperoptimization algorithm among a selection of both saturating and non-saturating activation functions.

Table 1 The hyperparameter configuration selected using the k-folds hyperoptimization and used to perform the "feature scaling" fits presented in this paper

Parameter	Flavor basis
Architecture	1-59-49-48-42-8
Activation function	$\|x\|\tanh(x)$
Initializer	`glorot_normal`
Optimizer	`Nadam`
Clipnorm	1.5×10^{-5}
Learning rate	4.3×10^{-3}
Maximum # epochs	19×10^{3}
Stopping patience	24% of max epochs
Initial positivity $\Lambda^{(\mathrm{pos})}$	34
Initial integrability $\Lambda^{(\mathrm{int})}$	10
N_{int}	40

Having identified the best settings for the hyperparameters, we can analyze the effect that changing the parametrization has on the PDFs and the predictions made with them. the χ^2 values obtained with the updated methodology are shown in Fig. 5 where they are compared to those of NNPDF4.0. From this it is clear that the feature scaling methodology provides a fit to the data as good as NNPDF4.0.

In what follows we will study the implications of the methodology in more detail, in many cases by comparing it to a PDF based on the same experimental dataset and theory setting, but produced using the NNPDF4.0 methodology. Specifically, we will perform various tests to validate the PDFs both in the extrapolation regions (see Sects. 5.2 and 5.3), as well as the data region (see Sect. 5.4). These test comprise the validation of the NNPDF4.0 methodology, and we will show that the performance of feature scaling is very similar to that of NNPDF4.0.

5.2 Validation of small-x extrapolation

To begin with, we need the PDFs to accurately describe the kinematic domain from which the methodology has not seen data during training. If we are able to determine the χ^2 for this

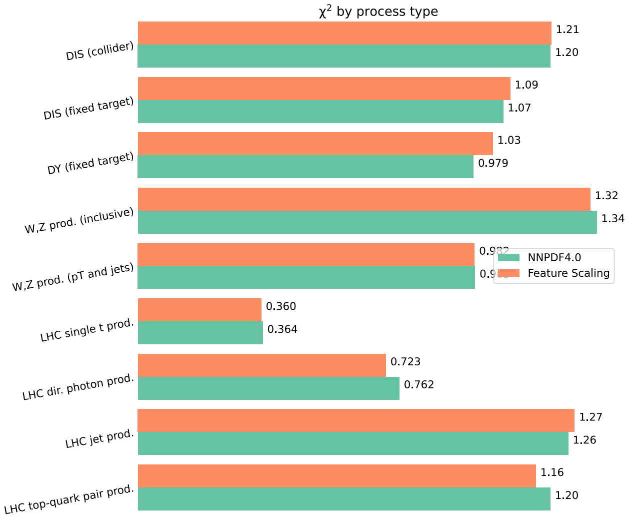

Fig. 5 A comparison of the χ^2 per process type between NNPDF4.0 (green) and feature scaling (orange), the total χ^2 of feature scaling is 1.17 while that of NNPDF is 1.16

unseen data, that would provide some insight into the generalization of our methodology in the extrapolation region.

By definition, testing the accuracy in a region where there is no data to test against is impossible. Given that waiting for a future collider to become operational could take decades, the next best thing we can do is to perform a fit to a “historic” dataset representing the knowledge available at an earlier point in time. To this end we utilize the “future test” technique introduced in Ref. [28], and used to validate the small-x extrapolation region of the NNPDF4.0 PDFs in Ref. [12]. For consistency we keep the same datasets as presented in the original future test paper (pre-HERA and pre-LHC). In short, the test goes as follows: if the prediction from our methodology is able to accommodate (within uncertainties) currently available data that was not included in the fit, then the test is successful and we consider the generated uncertainties to be faithful.

Since the aim of doing a future test is to determine the ability of a methodology for PDF determination to provide a generalized fit, we need to take into account not only the uncertainty of the experimental data but also the uncertainty of the PDF itself. This is done by redefining the covariance matrix in (6) as

$$C_{ij} = C_{ij}^{(\mathrm{exp})} + C_{ij}^{(\mathrm{pdf})}, \tag{9}$$

where $C_{ij}^{(\mathrm{pdf})}$ corresponds to the covariance matrix of the observables calculated from PDF predictions. Specifically, $C_{ij}^{(\mathrm{pdf})}$ is defined as

$$C_{ij}^{(\mathrm{pdf})} = \langle \mathcal{F}_i \mathcal{F}_j \rangle_{\mathrm{rep}} - \langle \mathcal{F}_i \rangle_{\mathrm{rep}} \langle \mathcal{F}_j \rangle_{\mathrm{rep}}, \tag{10}$$

where $\mathcal{F}_i^k$ is the prediction of the i-th datapoint using the k-th PDF replica with the average defined over replicas.

As can be seen in Fig. 6, where we compare the gluon and u quark PDFs of the NNPDF4.0 fit, to a PDF generated using the feature scaling methodology, the plots show good agreement between the two PDFs. While only the two partons are shown, this is representative of all flavors. The prediction of the feature scaling methodology in the extrapolation region is validated by performing a future test of the feature scaling methodology as has been done before for the NNPDF4.0 methodology in Ref. [11]. The results of this future test results shown in Table 2. Each column corresponds to a fit perform using all previous datasets (for instance, the pre-LHC fit includes all the data in pre-HERA as well). Instead, each row corresponds to the partial dataset used to compute the χ^2. We make a distinction between χ^2 inside parentheses with a covariance matrix as defined in (7), and the χ^2 without parentheses corresponding to a covariance matrix as defined in (9). Before seeing these results one may wonder whether, because all datasets are sensitive to the same large-x region, the datasets are consistent and thus the test is trivial. The answer to this becomes clear by looking at the χ^2 values inside the parentheses which indicate that when the PDF uncertainties are not considered the fit quality is very poor for unseen data.

We can analyze the result starting on the third row corresponding to the NNPDF4.0 dataset. For the fit that included the entire dataset (third column) it makes virtually no difference whether or not the PDF uncertainties are taken into account. This is quite different for the pre-HERA fit (first column): even though the central PDF is off ($\chi^2 = 7.23$), once its uncertainties are considered, the quality of the fit is comparable to that of NNPDF4.0 with a χ^2 of 1.29 compared to 1.21.

In the second row instead the pre-LHC dataset is considered. Both the NNPDF4.0 and the pre-LHC fit, where the dataset is included, produce a trivially good χ^2 for their fitted data. When we compute the prediction using the pre-HERA fit instead the number is much worse. Once again, upon considering the PDF uncertainties, the number is of order one, though still significantly larger than the corresponding values in the fits with pre-LHC or NNPDF4.0 data. This suggests that qualitatively good agreement is obtained but stability upon changes to the dataset can still be improved. It should be noted that in all cases the methodology used has been hyperoptimized for the full NNPDF4.0 dataset.

If we compare these results as presented in Table 2 for the feature scaling methodology, to the results for the NNPDF4.0 methodology shown in Table 3, we observe much the same properties. Indeed, even in cases in which the out-of-sample χ^2 of (7) differs greatly between both methodologies, the results are compatible once the PDF uncertainties are considered. While a strict passing criterion for the future test, such as a specific threshold χ^2, has never been defined, it allows us to test whether, when PDF uncertainties are con-

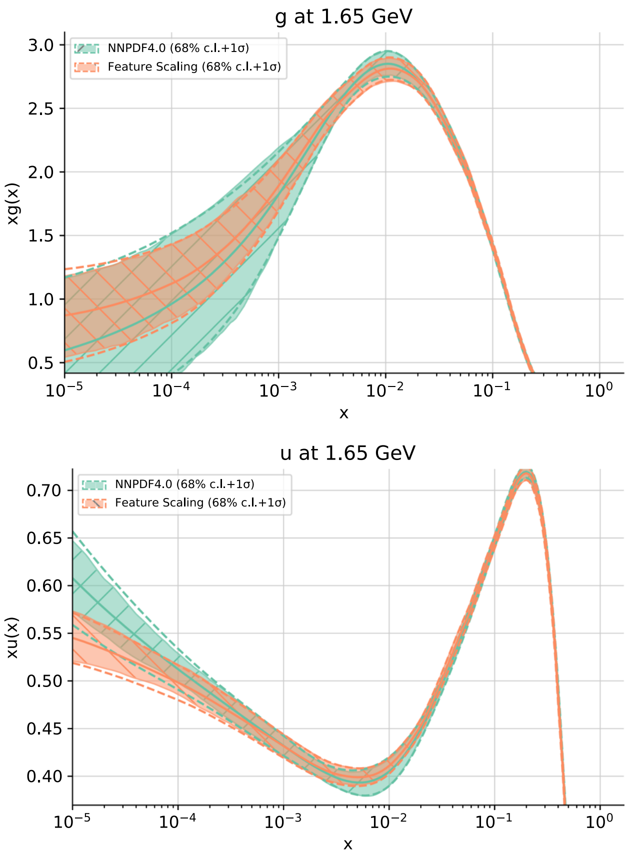

Fig. 6 Comparison of the gluon and u quark PDFs between a fit performed with the NNPDF4.0 methodology (green), and one with the feature scaling methodology (orange)

sidered, the agreement to out-of-sample data is of a similar level as that of fitted data where the PDF uncertainty is not considered. As said, this is indeed the case.

We must note however a deterioration of the results in Table 2 with respect to those of Table 3 which points to a greater dependence on the considered dataset with the feature scaling methodology. However, while the NNPDF4.0 set of PDFs has been finely tuned and highly optimized over many iterations and at high computational costs, the feature scaling methodology has not been subjected to the same degree of finetuning. It should also be noted that the NNPDF preprocessing ranges are determined using the full dataset while the aim of the feature scaling methodology that can directly accommodate different datasets. This could also explain why the out-of-sample χ^2 of feature scaling is actually better than that achieved by NNPDF4.0

5.3 Evaluation of large-x extrapolation

Upon removing the prefactor, we not only affect the small-x extrapolation region of the PDFs, but also the large-x extrapolation region. So far no test has been developed to test the validity in this range of the PDF domain. Using the future test to also test the faithfulness of the predictions in the large-x region cannot be done in the same way due to the limitations of the datasets that do not contain any large-x datapoints (irrespective of how we define large-x precisely). For example, removing all datasets which contain a point in $x \gtrsim 0.3$ leaves a set of datasets which do not provide sufficient constraints on the PDF to perform the future test. Nevertheless, here we will assess the large-x extrapolation behavior of the PDF produced with feature scaling.

To do so, let us have a look at the PDFs themselves in this region, and how the PDFs based on the NNPDF4.0 methodology compare to those that have been produced with feature scaling. A comparison of the gluon and strange PDF in the domain $0.6 < x < 1$ is shown in Fig. 7. Note here that there is no data available for $x > 0.75$, meaning that what is shown is mostly extrapolation region, and these representative examples show a good agreement between the NNPDF4.0 PDF and the feature scaling counterpart. We further want to point out that due to the lack of data in this region different, but all reasonable, parametrization choices can lead to very different results, as can be seen by comparing the NNPDF PDFs to those produced by MSHT or CT.

As a more rigorous check of the large-x extrapolation region one could create pseudodata based on predictions corresponding to PDFs that have a different (exponential) behavior in the extrapolation region, e.g. a change of the β_a exponent outside the data region. One can then perform a future test to this pseudodata, to quantify how well the PDFs generalize in the extrapolation region. The development of such a test, however, is well beyond the test of validity we provide here for the feature scaling methodology. Nevertheless, it can be an interesting check for a future release of PDF sets.

5.4 Validation of the data region

Where in Sect. 5.2 we performed a future test to validate the faithfulness of the PDFs in the extrapolation region where the PDFs are not constrained by data. Here, instead, we will validate the faithfulness of the PDFs in the data region by performing a closure test as first introduced in Ref. [13] and extended in the recent NNPDF4.0 paper. Below we repeat the closure test as performed in Sect. 6.1 of Ref. [11] for the feature scaling methodology, and unless stated otherwise, the same settings are used.

When fitting experimental data we are subject to complexities in the data such as inconsistencies between datasets or limitations of the theoretical calculations. These complexities make it more difficult to assess the performance of a fitting methodology by analyzing the result of a fit to experimental data. This realization is what led to the idea of a closure test, where, instead of fitting to experimental data, a

Table 2 χ^2 values per datapoint as obtained during a future test of the feature scaling methodology. The columns correspond to fits based on a given dataset, while the rows correspond to the datasets for which the χ^2 values are shown. While for the fit the dataset are inclusive (i.e., the NNPDF4.0 fit includes also the pre-LHC and pre-HERA datasets) the χ^2 is computed in an exclusive manner (i.e., the χ^2 as calculated for the NNPDF4.0 dataset only uses "post-LHC" data). The values in bold represent the performance on datasets that were not part of the training. The values inside parentheses correspond to a χ^2 defined with σ as defined in (9), while those without parenthesis are defined with only the experimental covariance matrix of (7)

Dataset	N_{dat}	Pre-HERA fit	Pre-LHC fit	NNPDF4.0 fit
Pre-HERA	2076	0.87 (0.92)	0.91 (1.03)	0.98 (1.08)
Pre-LHC	1273	**1.35 (5.61)**	1.17 (1.27)	1.18 (1.20)
NNPDF4.0	1269	**1.29 (7.23)**	**1.22 (4.72)**	1.21 (1.29)

Table 3 Same as Table 2 for the NNPDF4.0 methodology

Dataset	N_{dat}	Pre-HERA fit	Pre-LHC fit	NNPDF4.0 fit
Pre-HERA	2076	0.87 (0.91)	0.94 (1.01)	1.01 (1.06)
Pre-LHC	1273	**1.22 (26.1)**	1.18 (1.21)	1.17 (1.20)
NNPDF4.0	1269	**1.28 (22.6)**	**1.28 (2.15)**	1.23 (1.29)

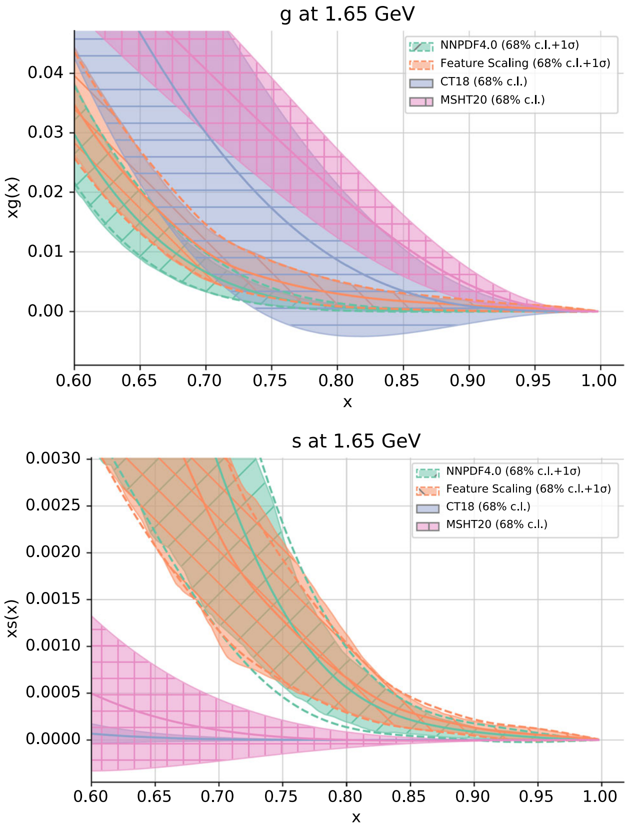

Fig. 7 Comparison of the large-x extrapolation regions of the gluon (top) and the strange (bottom) PDFs between NNPDF4.0 (green), feature scaling (orange), CT18 [15] (blue), and MSHT20 [14] (pink)

fit to pseudodata is performed. This pseudodata is generated by taking a fitted PDF as input, and from that calculating the observables corresponding to those in the experimental datasets, thereby creating a dataset with an associated, and known, underlying PDF. This allows us to test whether our methodology is able to faithfully reproduce the underlying PDF. To test whether our methodology was successful, a number of statistical estimators are considered that we will discuss next. For a detailed motivation of these estimators we refer the reader to Ref. [11]. As underlying truth we use one non-central replica from a feature scaling fit.

A first statistical estimator to consider is the Δ_{χ^2}

$$\Delta_{\chi^2} = \chi^2[f^{(\text{cv})}] - \chi^2[f^{(\text{ul})}], \tag{11}$$

where $\chi^2[f^{(\text{cv})}]$ is the loss evaluated for the expectation value of the fitted model predictions, while $\chi^2[f^{(\text{ul})}]$ is the loss evaluated for the predictions of the PDF used as underlying law. This latter loss does not vanish, because the pseudodata includes a Gaussian random noise on top of the central value predictions made using the underlying law. As such, Δ_{χ^2} can be understood as an indicator for overfitting or underfitting: if $\Delta_{\chi^2} > 0$, that indicates underfitting, while $\Delta_{\chi^2} < 0$ indicates overfitting. For the feature scaling methodology, the average Δ_{χ^2} as evaluated over observables corresponding to the full NNPDF4.0 dataset is $\Delta_{\chi^2} = -0.002$ (compared to $\Delta_{\chi^2} = -0.009$ for NNPDF4.0), which is at the per mille level indicating a negligible amount of overfitting.

Now let us estimate the faithfulness of the PDF uncertainty at the level of observables. For this we use the bias over variance ratio as defined in Eq. (6.15) of Ref. [11]. Here "bias" can be understood as a measure of the fluctuations of the observable values with respect to the central value prediction of the fitted PDF, while "variance" can be understood as the fluctuations of the fitted PDF with respect to its central value prediction. Thus if the methodology has faithfully reproduced the uncertainties in the underlying data (bias), this uncertainty should be equal to the uncertainty in the predictions of the PDFs (variance), and hence the bias to variance ratio R_{bv} is expected to be one.

To test this, the value of R_{bv} is determined for out-of-sample data. Specifically, we fit the PDFs to the NNPDF3.1-like dataset as defined in Ref. [11], and then evaluate the value of R_{bv} for the data that is part of the NNPDF4.0 dataset but has not already been included in the NNPDF3.1-like dataset. This allows us to test how well the predication made using a PDF fitted with a given methodology generalizes to unseen data. The value of the bias to variance ratio found for the new methodology is $R_{\mathrm{bv}} = 1.03 \pm 0.04$ (compared to $R_{\mathrm{bv}} = 1.03 \pm 0.05$ for NNPDF4.0), where again the uncertainty corresponds to a 1σ bootstrap error, meaning the agreement to the expected value of $R_{\mathrm{bv}} = 1$ is at the 1σ level.

To estimate the faithfulness of the PDF uncertainty at the level of the PDF we calculate a quantile estimator in PDF space $\xi^{(\mathrm{pdf})}_{1\sigma}$. This quantity corresponds to the number of fits for which the 1σ uncertainty band covers the PDF used as underlying law. This is determined for fits performed to pseudodata covering the full NNPDF4.0 dataset. The result is $\xi^{(\mathrm{pdf})}_{1\sigma} = 0.70 \pm 0.02$ (compared to $\xi^{(\mathrm{pdf})}_{1\sigma} = 0.71 \pm 0.02$ for NNPDF4.0), where the uncertainty is a 1σ uncertainty determined through bootstrapping [29,30]. Thus the observed $\xi^{(\mathrm{pdf})}_{1\sigma}$ value is in agreement with the expected value of 0.68 within 1σ.

An analogous estimator can be calculated for the theory predictions in data space as opposed to PDF space, providing a generalization to quantile statics of the bias of variance ratio R_{bv}. Similar to the bias over variance ratio, also for this estimator the values are calculated on out-of-sample data, where the PDFs have been determined using NNPDF3.1-like data. The expected value of this quantile estimator depends on the bias over variance ratio is $\mathrm{erf}(R_{\mathrm{bv}}/\sqrt{2}) = 0.67 \pm 0.02$ (compared to $\mathrm{erf}(R_{\mathrm{bv}}/\sqrt{2}) = 0.67 \pm 0.03$ for NNPDF4.0), which is in agreement with the calculated value of $\xi^{(\mathrm{exp})}_{1\sigma} = 0.69 \pm 0.02$ (and $\xi^{(\mathrm{exp})}_{1\sigma} = 0.68 \pm 0.02$ for NNPDF4.0).

6 Conclusions

In this paper we propose a series of modifications to the PDF parametrization through the treatment of the input data with a model-agnostic approach. The result is a simplified fitting procedure which opens the door for further automatization.

The implications of this new approach have been studied in the context of the NNPDF fitting framework. We tested the resulting methodology by its own merit and compared it to the latest release of NNPDF. We found agreement not only within the data region but also in the extrapolation region (where the choice of the prefactor can potentially have an effect) which further confirms the resilience of the NNPDF methodology upon parametrization changes.

This opens up new possibilities for further development of the methodology. In particular, the hyperoptimization procedure for PDFs is very sensitive to statistical fluctuations. Removing randomly chosen exponents eliminates statistical noise and greatly reduce the number of iterations need to get the best model.

Furthermore, while the fixed scaling of the input data or the choice of the preprocessing ranges to be sampled need to be reassessed when the dataset changes considerably (especially if the extrapolation regions change as a consequence) the proposed *feature scaling methodology* can accommodate data changes automatically, becoming more robust as the amount of data increases.

Acknowledgements The authors thank Stefano Forte, Juan Rojo and Richard Ball for a critical reading of the manuscript and several useful comments, and Christopher Schwan for providing interpolation grids for cross sections for the validation of the resulting PDF fits at various stage of this work. This project is supported by the European Research Council under the European Unions Horizon 2020 research and innovation Programme Grant agreement number 740006.

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: The paper presents an novel model architecture for parton distribution function determination.]

References

1. J. Butterworth et al., J. Phys. G **43**, 023001 (2016). arXiv:1510.03865
2. K. Kovařík, P.M. Nadolsky, D.E. Soper, Rev. Mod. Phys. **92**, 045003 (2020). arXiv:1905.06957
3. R. Abdul Khalek, S. Bailey, J. Gao, L. Harland-Lang, J. Rojo, SciPost Phys. **7**, 051 (2019). arXiv:1906.10127
4. M. Constantinou et al., Prog. Part. Nucl. Phys. **121**, 103908 (2021). arXiv:2006.08636
5. R.D. Ball, L. Del Debbio, S. Forte, A. Guffanti, J.I. Latorre, A. Piccione, J. Rojo, M. Ubiali (NNPDF), Nucl. Phys. B **809**, 1 (2009). arXiv:0808.1231 [Erratum: Nucl. Phys. B **816**, 293 (2009)]
6. G. Cybenko, Math. Control Signals Syst. **2**, 303 (1989)
7. S. Forte, L. Garrido, J.I. Latorre, A. Piccione, JHEP **05**, 062 (2002). arXiv:hep-ph/0204232

8. R.D. Ball, L. Del Debbio, S. Forte, A. Guffanti, J.I. Latorre, A. Piccione, J. Rojo, M. Ubiali (NNPDF), Nucl. Phys. B **823**, 195 (2009). arXiv:0906.1958
9. R.D. Ball et al. (NNPDF), Eur. Phys. J. C **77**, 663 (2017). arXiv:1706.00428
10. S. Carrazza, J. Cruz-Martinez, Eur. Phys. J. C **79**, 676 (2019). arXiv:1907.05075
11. R.D. Ball et al. (2021). arXiv:2109.02653
12. R.D. Ball et al. (NNPDF) (2021). arXiv:2109.02671
13. R.D. Ball et al. (NNPDF), JHEP **04**, 040 (2015). arXiv:1410.8849
14. S. Bailey, T. Cridge, L.A. Harland-Lang, A.D. Martin, R.S. Thorne, Eur. Phys. J. C **81**, 341 (2021). arXiv:2012.04684
15. T.J. Hou et al., Phys. Rev. D **103**, 014013 (2021). arXiv:1912.10053
16. S. Alekhin, J. Blümlein, S. Moch, R. Placakyte, Phys. Rev. D **96**, 014011 (2017). arXiv:1701.05838
17. S.J. Brodsky, G.R. Farrar, Phys. Rev. Lett. **31**, 1153 (1973)
18. H.D.I. Abarbanel, M.L. Goldberger, S.B. Treiman, Phys. Rev. Lett. **22**, 500 (1969)
19. S. Forte, S. Carrazza (2020). arXiv:2008.12305
20. R.D. Ball, E.R. Nocera, J. Rojo, Eur. Phys. J. C **76**, 383 (2016). arXiv:1604.00024
21. V. Bertone, S. Carrazza, J. Rojo, Comput. Phys. Commun. **185**, 1647 (2014). arXiv:1310.1394
22. V. Bertone, S. Carrazza, N.P. Hartland, Comput. Phys. Commun. **212**, 205 (2017). arXiv:1605.02070
23. R.D. Ball et al., JHEP **04**, 125 (2013). arXiv:1211.5142
24. Z. Kassabov, Reportengine: a framework for declarative data analysis (2019). https://doi.org/10.5281/zenodo.2571601
25. A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, G. Watt, Eur. Phys. J. C **75**, 132 (2015). arXiv:1412.7420
26. F.N. Fritsch, R.E. Carlson, SIAM J. Numer. Anal. **17**, 238 (1980)
27. V. Bertone, S. Carrazza, N.P. Hartland, E.R. Nocera, J. Rojo (NNPDF), Eur. Phys. J. C **77**, 516 (2017). arXiv:1706.07049
28. J. Cruz-Martinez, S. Forte, E.R. Nocera, Acta Phys. Pol. B **52**, 243 (2021). arXiv:2103.08606
29. B. Efron, Ann. Stat. **7**, 1 (1979)
30. B. Efron, R. Tibshirani, Stat. Sci. **57**, 54 (1986)