UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

Corso di Laurea Triennale in Fisica

STABILITY STUDIES IN THE DETERMINATION OF PARTON DISTRIBUTIONS

Relatore:
Stefano Forte

Correlatore:
Juan Cruz-Martinez

Candidato:
Federico Settimo
Matricola n. 885209

Anno accademico 2018-2019

Abstract

The main aim of this thesis is to study the stability and evolution of NNPDF fits and to show the importance of stopping point, cross-validation and the possibility of overfitting of a neural network.

After a brief introduction, in which we explain what is a parton distribution, we analize the behavior of the fits in different scenarios. First, we analize the quality of the fit when stopped before the optimal stopping point and its stability, by evaluating the fits at different epochs. Then, we analize how the fitness of PDF may vary depending on cross-validation, evaluating its importance to avoid overfitting. At last, we evaluate also the effect of the positivity constrain on the fits.

Contents

Parton Distribution Functions

Parton Distribution Functions (PDFs) describe the substructure of hadrons in terms of partons: quarks and gluons. They are a fundamental tool to compute the processes at hadron colliders, such as LHC. They have such an important role because they allow us to make predictions and calculate cross sections of scattering processes.

1.1 Quantum Chromodynamics

Quarks and gluons are the fundamental constituents of hadrons. Theirs structure can be expressed in terms of the basic fields, the degrees of freedom of the theory describing their interaction: quantum chromodynamics (QCD). There is one field for each of the 6 quark (plus their anti-quark), and one for gluons, the gauge boson of strong force. Quarks also have an internal degree of freedom, the color. This internal degree of freedom is analogous to electric charge, but each quark can have three possible states: blue, red, and green. There exist 8 independent gluons.

At the leading order of QCD, we can say that a PDF $f_i(x, Q^2)$ represents the probability of finding an i type parton (quark or gluon) carrying a fraction x of the proton momentum at a scale Q^2.

1.2 Factorization

Factorization is a fundamental property of QCD. It allows us to divide cross section computation in two separate parts: a process-dependent parton cross section and a set of universal parton distribution functions. Thanks to this universality it is possible to determine PDFs from a particular process and than use the PDFs to obtain prediction for different processes. For example,

factorization allows to express the cross section for hadroproduction at a scale M_X as [1]:

$$\begin{aligned}\sigma_X(s, M_X^2) &= \sum_{a,b} \int_{x_{min}}^{1} dx_1 dx_2 f_{a/h_1}(x_1, M_X^2) f_{b/h_2}(x_2, M_X^2) \hat{\sigma}_{ab\to X}(x_1 x_2 s, M_X^2) \\ &= \sum_{a,b} \sigma_{ab}^0 \int_{\tau}^{1} \frac{dx_1}{x_1} \int_{\tau/x_1}^{1} \frac{dx_2}{x_2} f_{a/h_1}(x_1, M_X^2) f_{b/h_2}(x_2, M_X^2) C_{ab}(\frac{\tau}{x_1 x_2}, \alpha_s(M_X^2)),\end{aligned} \tag{1.1}$$

where s is the center-of-mass energy, $f_{a/h_i}(x_i, M_X^2)$ is the distribution of type a parton in the i-th incoming hadron, $\hat{\sigma}_{ab\to X}$ is the parton-level cross section for production of X, $x_{min} = \tau \coloneqq \frac{M_X^2}{s}$, and the hard coefficient function C_{ab} is a function of the scale M_X^2 and the dimensionless ratio of the scale to the center-of-mass energy $\hat{s}$ of the partonic subprocess $z = \frac{M_X^2}{\hat{s}} = \frac{\tau}{x_1 x_2}$.

For electroproduction, Equation1.1 is replaced by a factorized expression for the structure functions $F_i(x, Q^2)$ that parametrize the DIS cross section:

$$\frac{d^2\sigma^{NC,\ell^\pm}}{dx dQ^2}(x, y, Q^2) = \frac{2\pi\alpha^2}{xQ^4}\left[Y_+ F_2^{NC}(x, Q^2) \mp Y_- x F_3^{NC}(x, Q^2) - y^2 F_L^{NC}(x, Q^2)\right] \tag{1.2}$$

for neutral-current (NC) charged-lepton ($\ell^\pm$), where $F_L(x, Q^2) = F_2(x, Q^2) - 2xF_1(x, Q^2)$ is the longitudinal structure function, $Y_\pm = 1 \pm (1-y)^2$, written in terms of $y = \frac{p\cdot q}{p\cdot k} = \frac{Q^2}{xs}$, the electron momentum, where p and k are the incoming proton and lepton momentum and q ($q^2 = -Q^2$) is the transferred momentum, and the last step holds if we neglet the proton mass, where s is the center-of-mass energy. The factorized expression for the structure function is

$$F_i(x, Q^2) = x \sum_a \int_x^1 \frac{dz}{z} C_{i,a}\big(\frac{x}{z}, \alpha_s(Q^2)\big) f_a(z, Q^2), \tag{1.3}$$

where $x = \frac{Q^2}{2p\cdot q}$, the standard Bjorken variable, $C_{i,a}$ is the structure function computed with an incoming parton and f_a is the distribution of the parton a in the incoming hadron.

Equations 1.1 and 1.2 hold at their respective scale, M_X^2 or Q^2, but we can relate PDFs at different scales by perturbative evolution equations

$$\frac{\partial}{\partial \ln Q^2}\begin{pmatrix}\Sigma(x,Q^2)\\ g(x,Q^2)\end{pmatrix} = \int_x^1 \frac{dy}{y}\begin{pmatrix}P_{qq}^S(\frac{x}{y},\alpha_S(Q^2)) & 2n_f P_{qg}^S(\frac{x}{y},\alpha_S(Q^2))\\ P_{gq}^S(\frac{x}{y},\alpha_S(Q^2)) & P_{gg}^S(\frac{x}{y},\alpha_S(Q^2))\end{pmatrix}\begin{pmatrix}\Sigma(x,Q^2)\\ g(x,Q^2)\end{pmatrix} \tag{1.4}$$

$$\frac{\partial}{\partial \ln Q^2} q_{ij}^{NS}(x,Q^2) = \int_x^1 \frac{dy}{y} P_{ij}^{NS}(\frac{x}{y},\alpha_S(Q^2)) q_{ij}^{NS}(y,Q^2), \tag{1.5}$$

where g is the gluon distribution, Σ is the singlet quark distribution

$$\Sigma(x,Q^2) \coloneqq \sum_{i=1}^{n_f}\big(q_i(x,Q^2)+\bar{q}_i(x,Q^2)\big) \tag{1.6}$$

and the nonsinglet distributions are $q_{ij}^{NS}(x,Q^2) = q_i(x,Q^2) - \bar{q}_j(x,Q^2)$, any linearly independent set of $2n_f - 1$ differences of the quark and antiquark distribution. Perturbative evolution has some constrains due to conservation laws: the conservation of baryon number

$$\int_0^1 dx\big(q_i(x,Q^2) - \bar{q}_i(x,Q^2)\big) = n_i, \tag{1.7}$$

where $n_u = 2, n_d = 1, n_{s,c,b,t} = 0$, and the conservation of the total energy-momentum

$$\int_0^1 dx x\left[\sum_{i=1}^{n_f}\big(q_i(x,Q^2)+\bar{q}_i(x,Q^2)\big) + g(x,Q^2)\right] = 1. \tag{1.8}$$

Therefore, combining factorized expression in Equations 1.1 and 1.2 with the solution of Equations 1.4 and 1.5, it is possible to write the physical observables as the convolution of a prefactor with the PDFs at a reference scale.

1.3 PDF parametrizations

A set of PDFs is a set of functions for each $0 < x < 1$ at some reference scale Q_0.

In principle there are 13 independent PDFs (12 for quarks and antiquarks and one for the gluon), but, in practice, charm and heavier quarks are not independently determined but assumed to be generated only by QCD radiation. In most cases it is more convenient to express the PDFs of the six

light-quark as suitable linear combinations, such as the singlet combination of Equation 1.6.

A standard choice for of a PDF parametrization is made by assuming that

$$f_i(x, Q^2) = x^{\alpha_i}(1-x)^{\beta_i} g_i(x), \tag{1.9}$$

with g_i that tends to a constant for $x \to 0, 1$, with this choice motivated by the expectation that PDFs behave as power of x for $x \to 0$, due to Regge theory, and as a power of $1-x$ for $x \to 1$, due to quark counting rules [2]. g_i can vary between the different possible approachs, with some common choices that are polinomial or exponential of polinomial in x or $\sqrt{x}$. Tipically, these PDF sets are parametrized by $\approx 20 - 30$ parameters. The choice of a parametrization corresponds to projecting the infinite-dimensional problem of determining a PDF set of functions onto the finite-dimensional space of the parameters. This way, errors on PDFs are just error ellipsoids on in the parameters space.

An alternative choice can be to parametrize PDFs with a general functional form that doesn't include any theoretical prejudice. With this unbiased PDF choice, the absolute minimum of the figure of merit (like χ^2, described in Section 1.4) is not necessarily the best possible fit, because it can correspond to a result describing also random fluctuations (like the PDFs described in Section 2.2.1), therefore a systematic study is necessary in order to avoid this problems. With unbiased PDFs random fluctuations are smoothened out because the parametrization is not flexible enough.

The fact that the best fit is not uniquely defined shows that the problem of determining a set of functions from a finite set of experimental data is not mathematically well defined: it would mean obtaining infinite information from a finite set of data. Hence, theoretical assumptions, such as parametrization, are necessary.

1.4 Fitting methodology

The goodness of a PDF set is measured by minimizing a suitable figure of merit, defined as

$$\chi^2 = \sum_{i=1}^{N_{dat}} \sum_{j=1}^{N_{dat}} (D_i - T_i)(V^{-1})_{ij}(D_j - T_j), \tag{1.10}$$

where T_i are the theoretical predictions, D_i the data point, and

$$V_{ij} = \delta_{ij}(\sigma_i^{uncorr})^2 + \sum_{k=1}^{N_{corr}} \sigma_{k,i}^{corr}\sigma_{k,j}^{corr} \tag{1.11}$$

is the experimental covariance matrix, where every i-th data point ($i = 1, ..., N_{dat}$) is affected by uncorrelated uncertainty σ_i^{uncorr} and correlated systematic uncertainty $\sigma_{k,i}^{corr}$ for $k = 1, ..., N_{corr}$.

It is possible to rewrite Equation 1.10 by adding N_{corr} shift parameters r_k as

$$\chi^2 = \sum_{i=1}^{N_{dat}} \left(\frac{\hat{D}_i - T_i}{\sigma_i^{uncorr}} \right)^2 + \sum_{i=1}^{N_{dat}} r_k^2, \tag{1.12}$$

where

$$\hat{D}_i \coloneqq D_i - \sum_{k=1}^{N_{corr}} r_k \sigma_{k,i}^{corr}. \tag{1.13}$$

A confidence interval in the space of PDFs is determined by minimizing a suitable measure of goodness of fit, which is not trivial because it requires the definition of a probability measure on a space of functions [3]. One first method of representing probability distributions in PDFs space is the Hessian method, based on the standard least-squares method [4]. This method is based on the assumptions that the parameter's probability distribution is a multi-gaussian, determining a central PDF as the one that minimize χ^2 and determining a 1-σ confidence level as the volume enclosed by the $\chi^2 = \chi^2_{min} + T^2$, where T is a tolerance parameter[1]. This method is usually used with PDFs using a relatively small number of parameters and one of its advantages is that it allows a compact representation and computation of PDF uncertainties by providing eigenvectors of the Hessian matrix rescaled by their respective eigenvalues. The best fit of any value $F(S)$ and its 1-σ uncertainty are

$$F_0 = F(S_0) \qquad \sigma_F = \sqrt{\sum_{i=1}^{N_{par}} \left[F(S_i) - F(S_0)\right]^2}, \tag{1.14}$$

where S_0 is the central set of PDFs and S_i, $i = 1, ..., N_{par}$, are the 1-σ error sets, corresponding to the variation of each eigenvector.

[1] With the standard choice $T^2 = 1$ the best fit parameter fluctuate much more than it would do if it actually provided a 1-σ confidence level

An alternative method is the Monte Carlo method, generating N_{rep} PDF sets S^k by assigning a Monte Carlo sample of PDF replicas, and this way it is possible to obtain the probability distribution of PDFs. The best fit is determined as expectation value, while the 1-σ interval is the standard deviation:

$$F_0 = \frac{1}{N_{rep}} \sum_{k=1}^{N_{rep}} F(S^k), \quad \sigma_F = \sqrt{\frac{1}{N_{rep}+q} \sum_{k=1}^{N_{rep}} \left[F(S^k) - F_0\right]^2}. \tag{1.15}$$

Monte Carlo method has the advantage that provides a direct representation of the probability distribution, without the need to make any assumption on the shape of the probability distribution of the parameters.

1.4.1 Cross-validation

If there is a very large number of parameters, determining the best fit could be non trivial, due to false minima and fluctuations. There are several ways to avoid this problem. One possible way is to add a penalty term to the χ^2 to the PDFs that are too complex, penalising the PDFs that are longer.

An alternative method is cross-validation [5]. This method consists in randomly dividing the data in two sets, training and validation, and computing their χ^2 separately, but only the training one (χ^2_{tr}) is minimized. At the beginning, both χ^2_{tr} and χ^2_{vl} decrease, but at some point (see Figure 1.1) training continues decreasing, while validation has a global minimum.

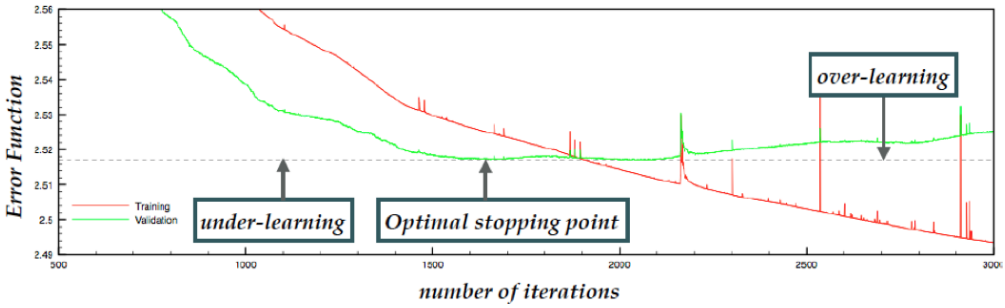

Figure 1.1: Training and validation χ^2 at different epochs

This point is the optimal stopping point: if the process is stopped before this point we have under-learning, while if it is stopped after this point we have over-learning: the PDF is trained to describe also statistical fluctuations.

When the experimental data are randomly divided between training and validation, it is usually with equal probability, but it's also possible to choose different probabilities, like what is done in Section 2.2, using only training.

1.5 Neural networks for determination of PDFs

The neural network (NN) approach [5, 6], used in this work, allows to avoid theoretical biases that can be incurred when particular functional forms are adopted.

The main idea of the use of neural networks is that they can be used as universal unbiased interpolators: starting from a Monte Carlo representation of the probability density of data points, they can be used to produce a representation of the probability density everywhere.

Obtaining a parametrization from the data requires two distinct steps [7]. In the first step artificial data are produced as N_{rep} replicas of the original set of N_{dat} data points, obtaining a Monte Carlo set of pseudodata

$$F_i^{(art)(k)} = \left(1 + r_N^{(k)}\sigma_N\right)\left(F_i^{(exp)} + \sum_{p=1}^{N_{sys}} r_p^{(k)}\sigma_{i,p} + r_i^{(k)}\sigma_{i,s}\right), \tag{1.16}$$

$k = 1, ..., N_{rep}$, $i = 1, ..., N_{dat}$, where F_i represents one single point, σ_N is the total normalization uncertainty,$\sigma_{i,p}$ are the N_{sys} correlated systematic errors, $\sigma_{i,c} = \sum_{p=1}^{N_{sys}} \sigma_{i,p}^2$ is the sum of all correlated systematics, and $r^{(k)}$ are independent univariate gaussian random numbers. The N_{rep} sets of data point are distributed as an N_{dat}-dimensional multi-gaussian with expectation value equal to the experimental value and standard deviation as the error of experimental points.

The second step is the interpolation between data points with neural networks. It consists on training N_{rep} sets of neural networks, each of them based on the data in one single replica. At the end, we have N_{rep} PDFs and from these value we can determine the mean value of the parton distribution for each x as the average over all the replicas, while the uncertainty is the variance of the values. This way it is possible to eliminate the problem of choosing a value of $\Delta\chi^2$ that corresponds to 1-σ contour.

Beyond leading order, PDFs do not need to be positive defined. However, the requirement for some measurable physical observables to be positive still imposes a generalized positivity constraint on the PDFs [8]. For example, an important theoretical constraint is the positivity of physical cross–section.

Therefore, positivity must be imposed on observable hadronic cross-sections and not on parton distributions, which do not necessarily need to be positive (except at leading order where the probabilistic interpretation holds) [9, 10]. The positivity constrain is imposed through Langrange multipliers, it consists in adding pseudo-data for cross sections with very small uncertainties, such that a negative cross section would lead to a very high contribution to the χ^2.

1.5.1 The NNPDF neural network

In NNPDF fits up to 3.1, a multi-layer feed-forward neural network is used, where each flavour is independent. This NN has a 2-5-3-1 architecture: it has four layers with, respectively, 2,5,3 and 1 neuron. The first layer is the input layer: it receives two inputs (x and $\ln 1/x$); while the last layer is the output layer, whose value is directly related to the value of the PDF at the scale Q_0. The output of the j-th neuron of the l-th layer $\xi_j^{(l)}$ ($j = 1, ..., n_l$, $l = 1, ..., 4$, where n_l is the number of neurons in the l-th layer) is given by an activation function $g(x)$:

$$\xi_j^{(l)} = g(h_j^{(l)}), \qquad g(x) = \frac{1}{1 + e^{-x}} \tag{1.17}$$

for the first three layers and $g(x) = x$ for the last layer, depending on a linear combination of the outputs of all neurons on the previous layer $\xi_j^{(l-1)}$,

$$h_j^{(l)} = \sum_{i=1}^{n_l - 1} \omega_{ji}^{(l)} \xi_i^{(l-1)} - \theta_j, \tag{1.18}$$

where the weights ω_{ij} and the thresholds θ_j are free parameters of the network, to be determined during the fitting. Up to NNPDF 3.1 training was done using genetic algorithms.

For the work presented in this thesis a new framework has been used, where multiple parameters can be tuned, thus allowing the studies that are presented in Chapter 2. The main differences of this new method is that it uses the same NN for all the flavours, using a bigger network with a 35-25-8 architecture. Furthermore, minimization is performed using a variant of gradient descend method, with the modifies on weights and thresholds are done in the opposite direction of the gradient of χ^2 $\left(\frac{\partial \chi^2}{\partial \omega_{ij}^{(l)}}, \frac{\partial \chi^2}{\partial \theta_i^{(l)}}\right)$, to determine the best fit parameters.

Results

2.1 Dependence on stopping point

The first part of this work is devoted to study the quality of the fit when it is stopped before the optimal stopping point.

The fit has a maximum number of epochs of 40000, to be sure that the neural network has enough time to reach its final state. We divide the total number of epochs in 200 steps, each of them of 200 epochs, in order to be able to monitor the fit in many different points. For each step we have 100 replicas.

By combining the data of each of the 200 steps, we also create a visual representation of the evolution of the fits, and this animation allowed us to chose the most significant steps, which are reported in this work to show the evolution of the fits.

2.1.1 The neural network at the end of the epochs

First of all, we must be sure that the PDFs generated from the neural network at the end of all the epochs are compatible with the current PDFs used for computations.

So we compare the PDF we have at the end (at step 200) with a reference PDF, with the results shown in Figure 2.1. In Figure 2.2 we show the distance between the fits, according to de definition given in [11], where a distance of 5 with 100 replicas correspond to a compatibility of 0.5 σ. Reference fit corresponds to the NNPDF 3.1 [12] methodology, while the current fit correspond to a new methodology [13]. All work in this thesis has been implemented with the new code described in [13]. The results show that the fits converge to a stable result, compatible with the reference fit, so it makes sense for us to compare its state at the end with its state at some points in the middle. The only significative differences between the two fits are in the

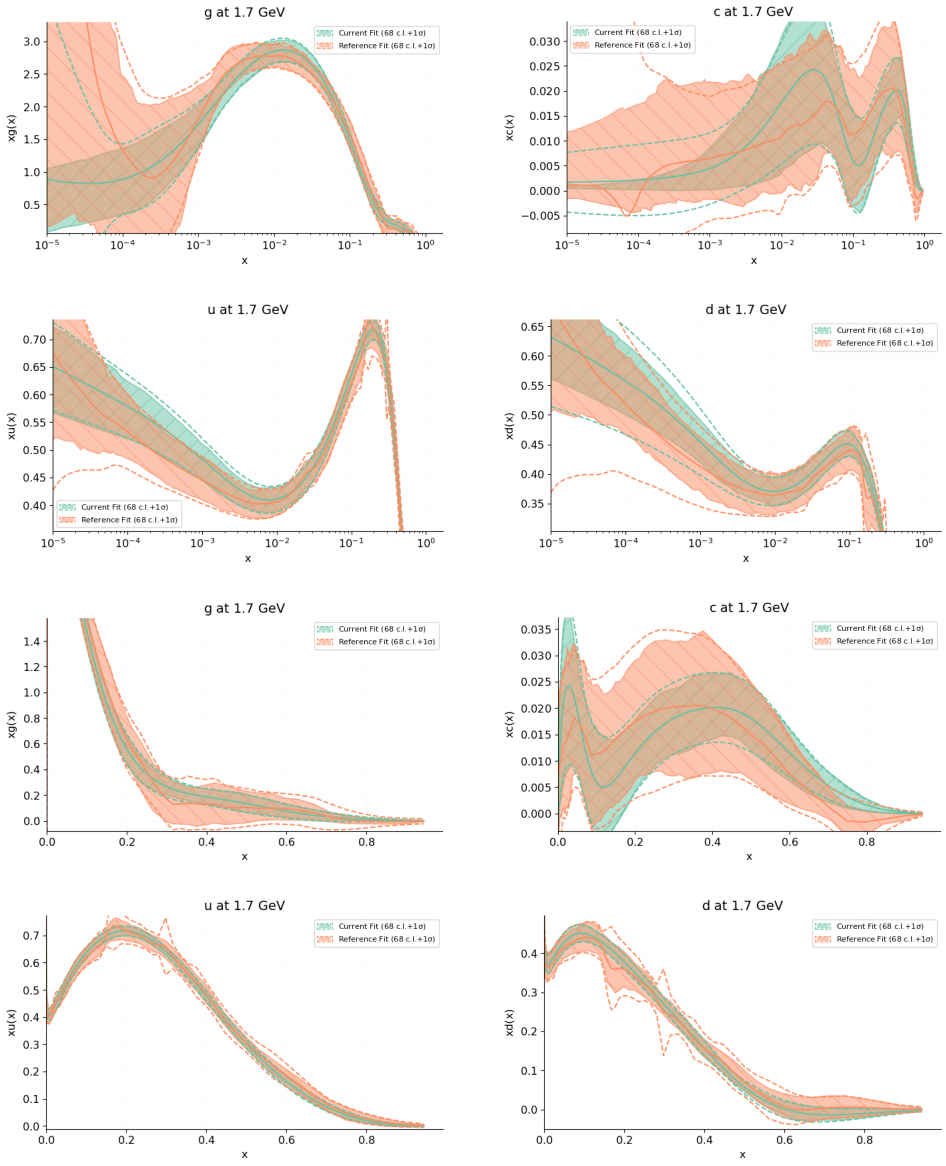

Figure 2.1: PDF at the end compared to a reference PDF

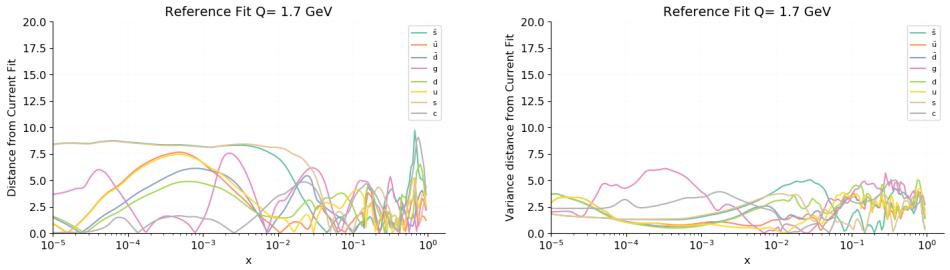

Figure 2.2: Distance between the PDFs and the reference fit (left) and variance distance (right)

low-x regime ($x < 10^{-4}$), which is also the slowest to converge to the final PDF.

2.1.2 Step 3 of 200 (600 epochs)

The first step we evaluated is after 600 epochs (step 3) compared to the fits at the end of all the epochs (step 200). As we can see in Figure 2.6 the plots are very different because at this point the neural network is very far from converging to a stable and correct model. Furthermore, the 1-σ contour is larger than the one for the fits at the end of the epochs. Despite the poor fits at step 3, we can observe that at high-x regime ($0.8 \leq x \leq 1$) the plot starts to resemble the reference one. Also, in Figure 2.5 we can see that the PDFs are very distant.

The χ^2 evaluated on experimental data also proves this. For the last replica we have $\chi^2 = 1.12059$, while at step 3 $\chi^2 = 3.22907$. For example, by looking at Figure 2.3, we can see that for each experiment considered the χ^2 at step 3 is way higher than the one at the end.

Also, if we compare the prediction of the two fits to the actual data, we can see, like in Figure 2.4, that the data at step 3 are far from both the actual data and the fit at the end of the epochs. So, the neural network and the PDFs generated from it are very far from describing well the underlying physics.

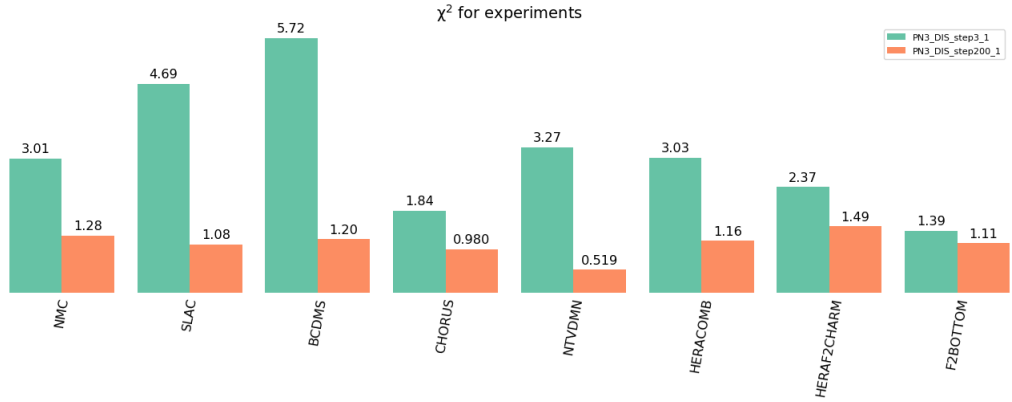

Figure 2.3: χ^2 by experiment at step 3

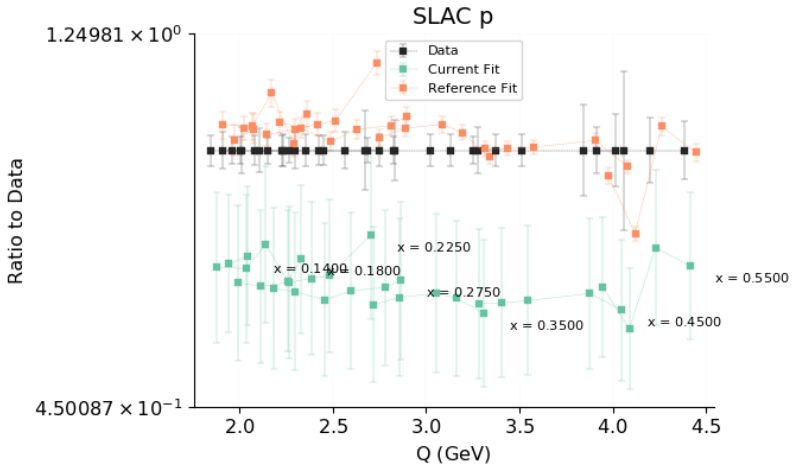

Figure 2.4: Data prediction at step 3 and at step 200

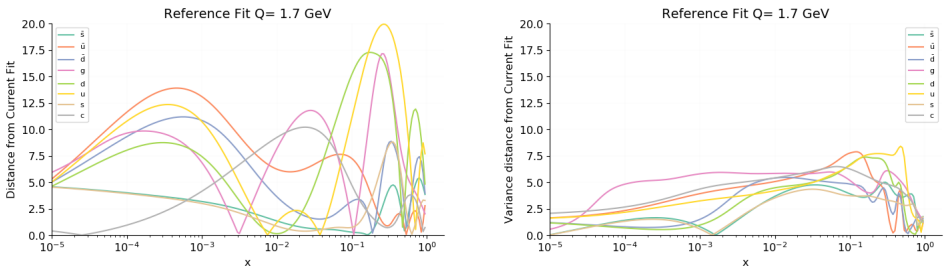

Figure 2.5: Distance between the PDFs at step 3 and the PDFs at step 200 (left) and variance distance (right)

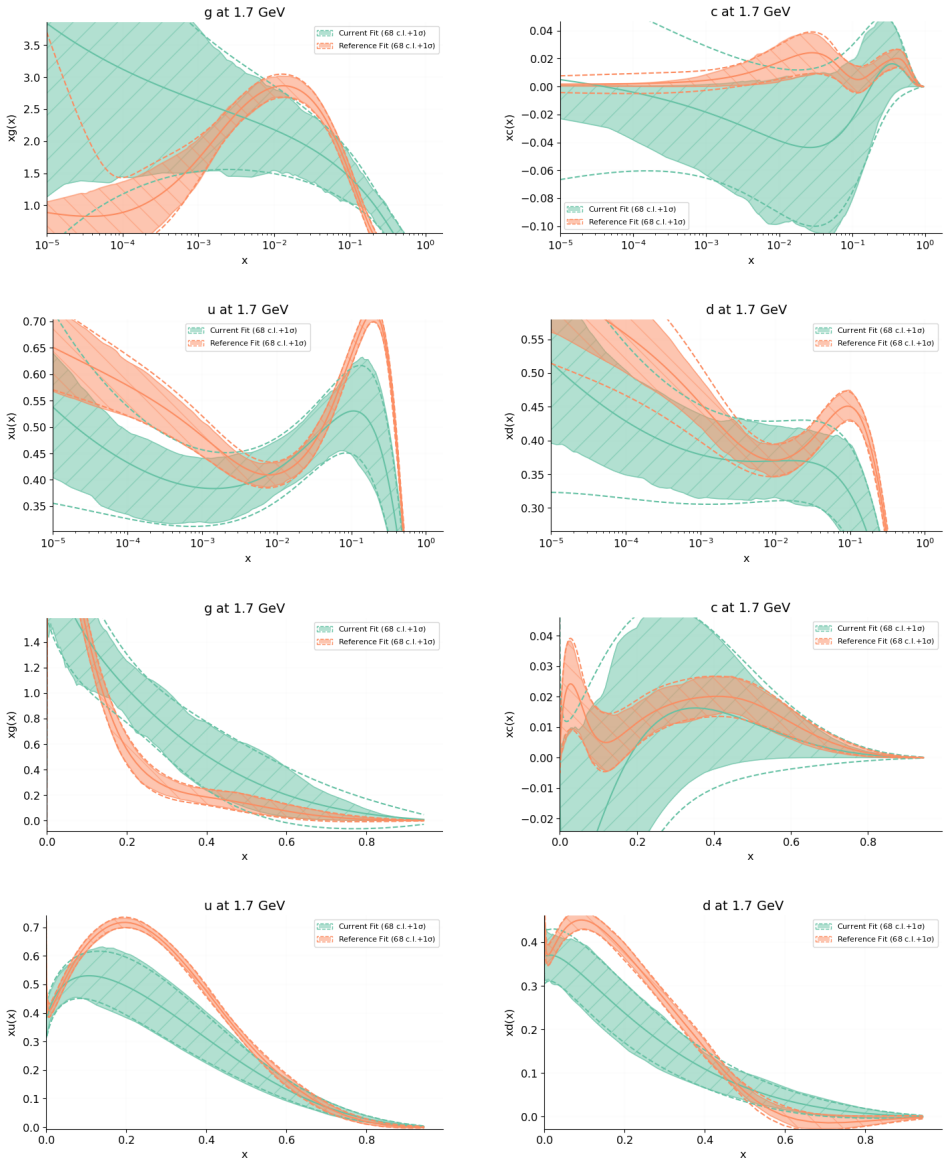

Figure 2.6: PDFs at step 3 compared to step 200

2.1.3 Step 5 of 200 (1000 epochs)

In Figure 2.7, we compare step 5 and step 200: considering the previous plots in Figure 2.6, step 5 and step 3 substantially don't differ from each other Furthermore, as we expect by looking at the plots, the χ^2 has't improved much: now we have $\chi^2 = 3.08714$.

Even if the first PDF converged to a quite good high-x regime description in only 600 epochs, from this fit we can conclude that, even if 600 epochs are enough to make a first fit, we need way more than the same amount of epochs to gain the fine tuning that actual fits require.

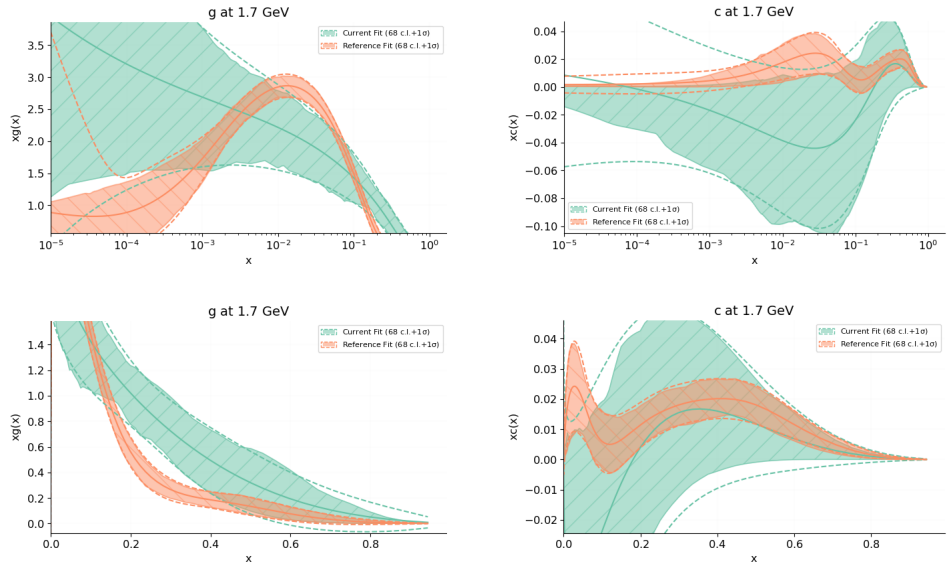

Figure 2.7: PDFs at step 5 compared to step 200

2.1.4 Step 15 of 200 (3000 epochs)

Around step 15, so after 3000 epochs, we start having the first significant improvements. First of all, the χ^2 improved, both the total χ^2 having a value of 2.48924, and the χ^2 per experiment, shown in Figure 2.8, with all the values improved compared to the ones of step 3 (Figure 2.3). In Figure 2.9, we can observe that most of the prediction are compatible with the data, within the error, even if the central values are still different, while at epoch 3 (Figure 2.4) almost none of the predicted data points were compatible.

Furthermore, by looking at the plots of Figure 2.11 we can also see an important improvement from step 5, also noticeable looking at the distances (Figure 2.10), but we can also observe that the PDF convergence differs in different kinematics regimes. We have an high-x regime ($x > 0.8$) of very fast convergence: here the NN already describes almost perfectly the PDF, after only 3000 epochs. On the other hand, we have the low-x regime where often the NN is very far from describing the data. Also, the 1-σ error is smaller than the steps considered previously.

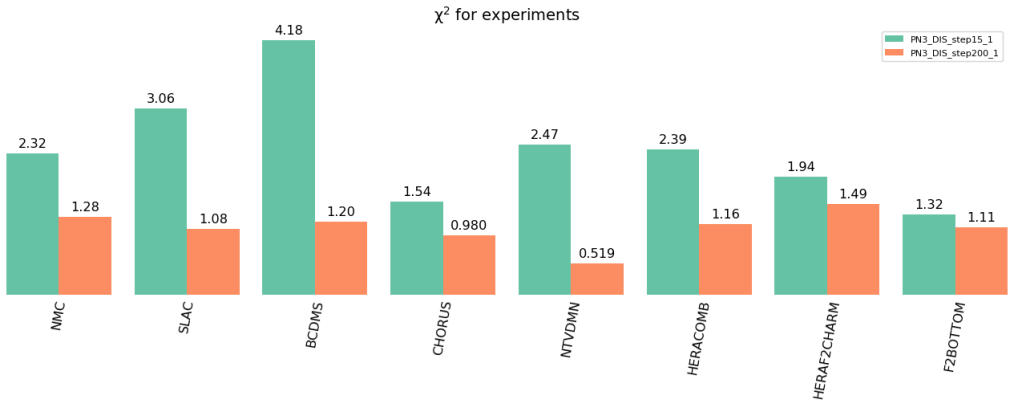

Figure 2.8: χ^2 by experiment at step 15

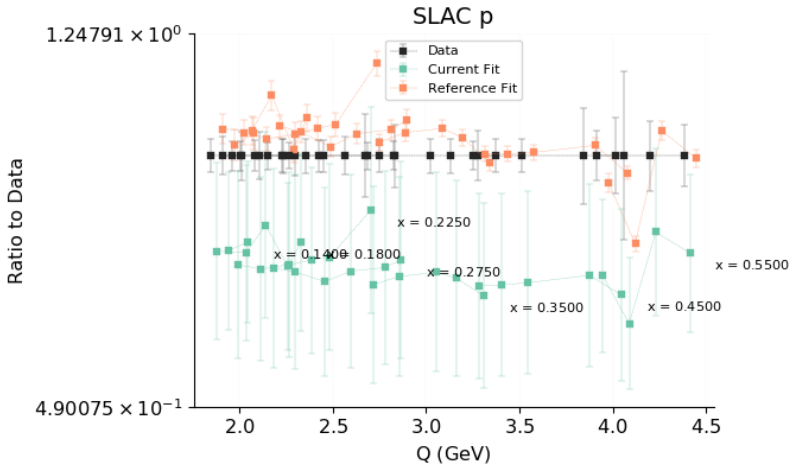

Figure 2.9: Data prediction at step 15 and at step 200

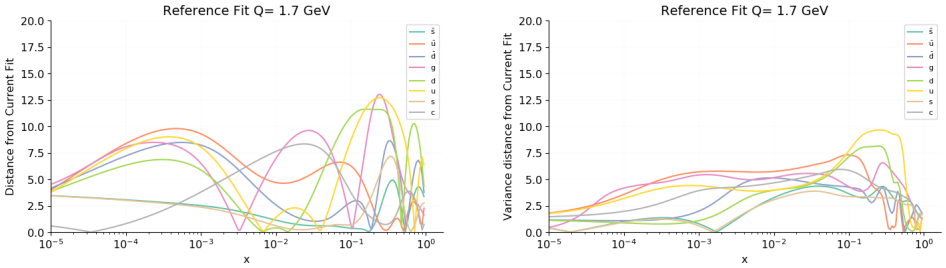

Figure 2.10: Distance between the PDFs at step 15 and the PDFs at step 200 (left) and variance distance (right)

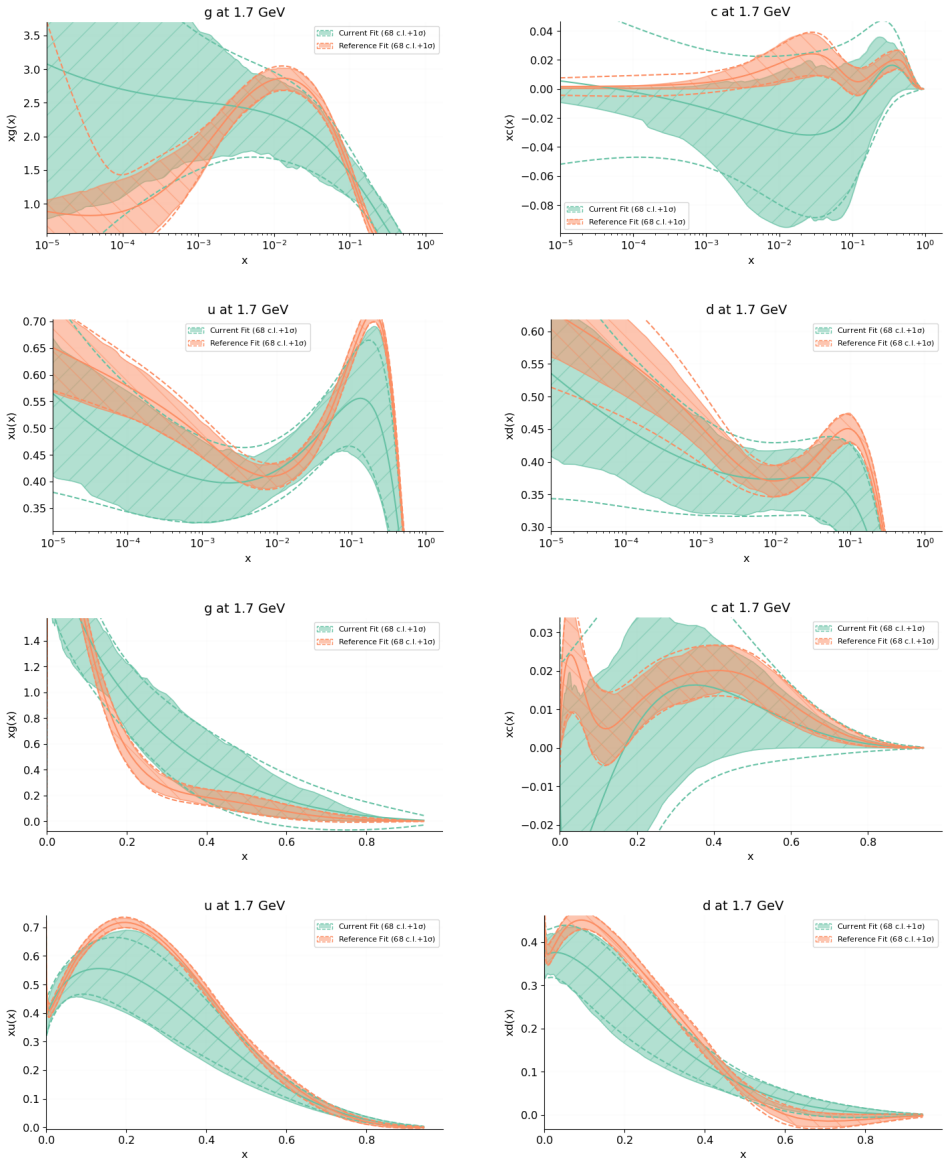

Figure 2.11: PDFs at step 15 compared to step 200

2.1.5 Step 25 of 200 (5000 epochs)

As we increase the number of the epochs, the quality of the fit improves. Here, as an example, we show the fit after 5000 epochs. First of all, we have a good value of $\chi^2 = 1.25982$. Then, looking at Figure 2.12, we can see that the central replica is almost everywhere inside the 1-σ contour of the reference fit. This is also seen in the χ^2: it takes less time to go from $\chi^2 > 100$ (at the first step) to $\chi^2 \approx 3$, than it takes to go from $\chi^2 \approx 3$ to $\chi^2 = 1.2$. Analogously, from the fits we see that it takes only 600 epochs to start having a reasonable shapes, but it needs 5000 epochs to have a compatible central value, but still with some significant difference is in the envelope, which is larger than the reference and somewhere non compatible, but for most of the values of x they are compatible. Analogously to the previous point, we can still observe that the main differences are at low-x. Also, the size of the 1-σ error is compatible to the error at the end, even if it is bigger at step 25.

Furthermore, by looking at the data predictions (Figure 2.13) and at the distances (Figure 2.14), we can notice that all the predictions are compatible and the distances are small, but still the fits are different.

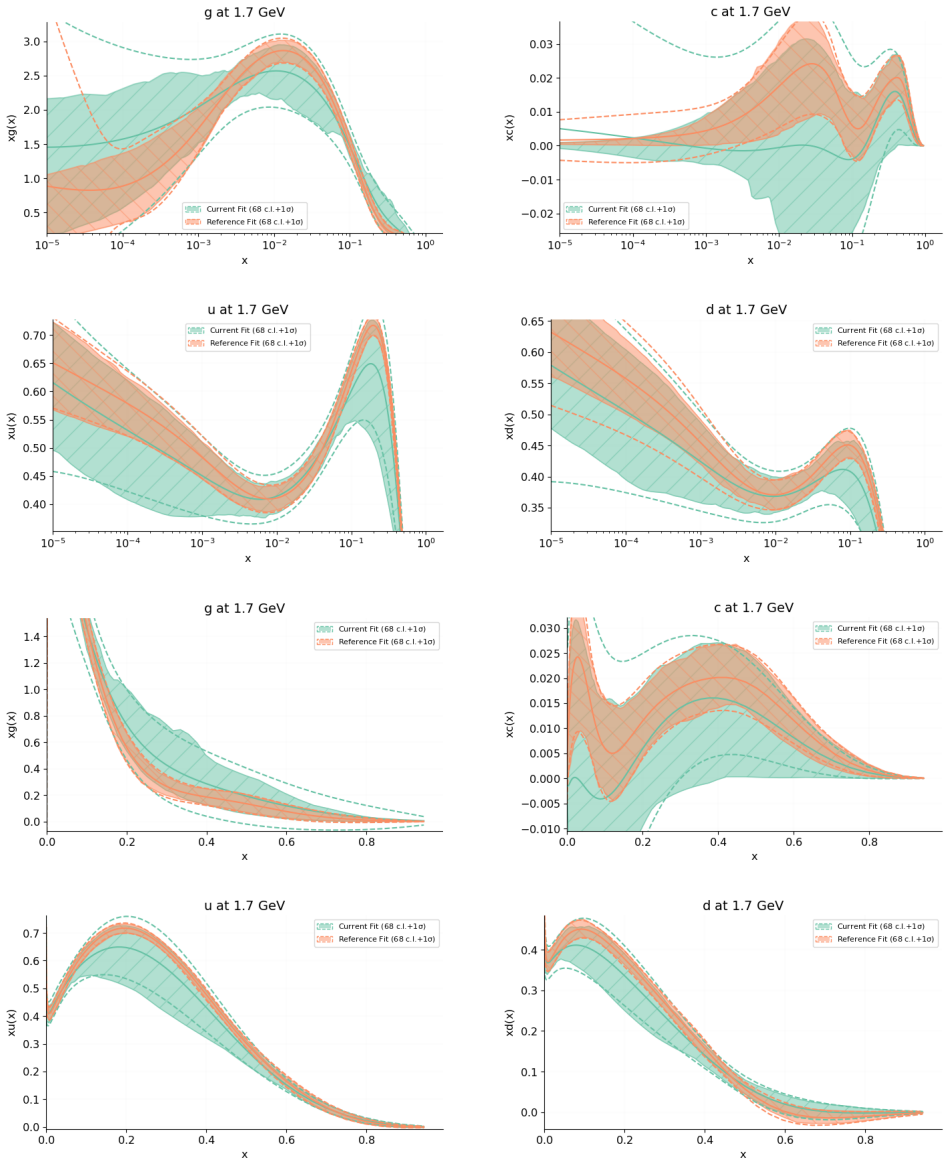

Figure 2.12: PDFs at step 25 compared to step 200

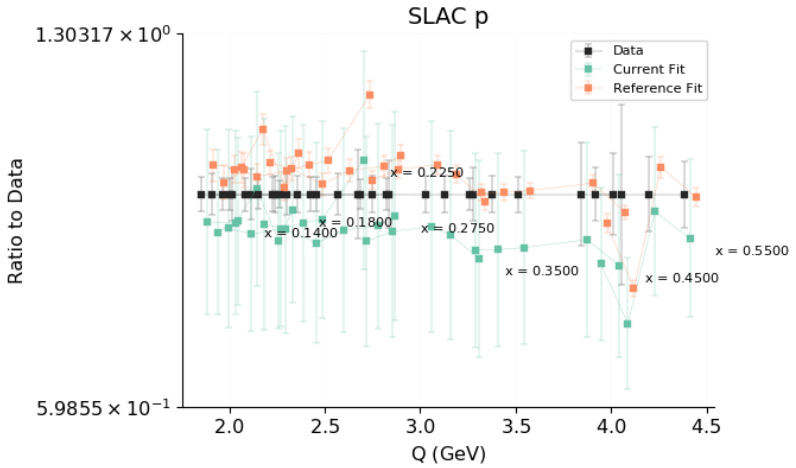

Figure 2.13: Data prediction at step 25 and at step 200

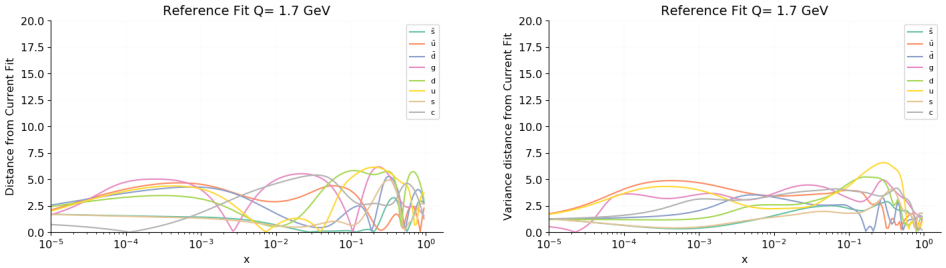

Figure 2.14: Distance between the PDFs at step 25 and the PDFs at step 200 (left) and variance distance (right)

2.1.6 Stability after step 25

After step 25 we can see that the NN converges to a state very similar to the final result, in particular from about step 50 the shapes of the fits and the values of χ^2 are the same of step 200. Furthermore, the χ^2 is at its minimum from step 50: the fits take more epochs to pass from $\chi^2 \approx 3$ to $\chi^2 = 1.2$ than to pass from $\chi^2 = 1.2$ to its final value of $\chi^2 = 1.1$. In the same way, after step 25, where the central values are compatible, we still need the same number of epochs more to converge to fits with also the same envelope.

In Figure 2.15 we can see that most of the replicas reach their final state before 10000 epochs, even if many replicas take a longer time. Therefore a suitable stopping point could be after about 15000 or 20000 epochs, in order to have the most of the replicas at theirs optimal stopping point, saving some important computation time with respect to a stopping point after 40000 epochs.

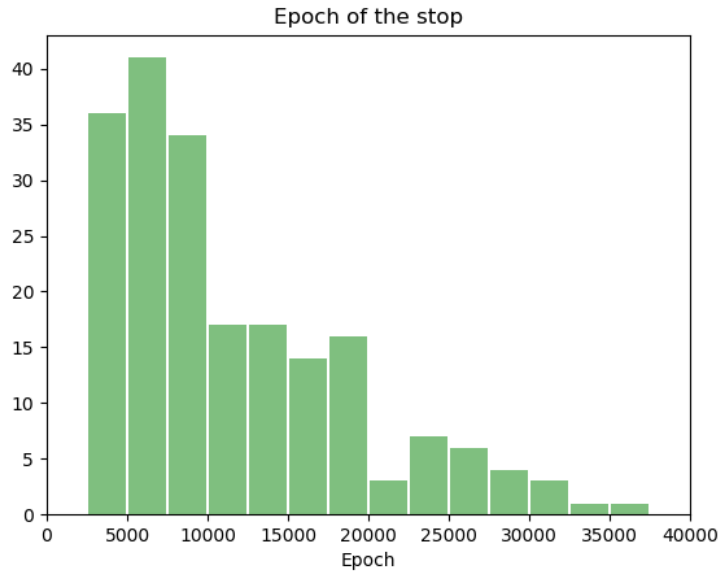

Figure 2.15: Stopping point of the different replicas

2.2 Dependence on cross-validation

Provided that the NN converges and after studying its evolution to this stable form, we now proceed to study how the NN changes depending on cross-validation.

Therefore, we run the same NN of Section 2.1, but using all the data as training and ignoring validation. As seen in Section 1.4.1, we expect that without cross-validation we have an overfitted network, describing also statistical fluctuations.

But actually, in Figure 2.16, we can see that the two fits are very similar, expect for some small difference for c quark. Also, the distance between the two fits (Figure 2.17) are very similar. Furthermore, the values of χ^2 (1.08 versus 1.12) and arclength[1] are very similar.

2.2.1 Comparison to an overfitted network

For reference, we show now an actually overfitted PDF, obtained by using a larger network, compared to the first one (Figure 2.18) and to the one without cross-validation (Figure 2.19).

Here we can see that overfitted PDFs are way more wobbly than the others, as an evidence of overfitting. Also, from arclengths, (Figure 2.20), we can see that the overfitted arclengths are way bigger than the other two, as we expect from overfitting. By comparing these PDFs, we can conclude that the one without cross-validation is way more similar to the first one at the end of the epochs than to the overfitted one. Therefore, cross-validation has a marginal role in preventing overfitting, even if we expected that it is the factor that stops overfitting from happening.

One possible explanation of the marginality of cross-validation is that the training data are enough to make a good fit also for validation data points, so the validation data are actually redundant: the huge amount of data points flattens out random fluctuations and the neural network is not big and flexible enough to describe these fluctuations, but instead describes an average of the fluctuations. We can see a proof of that by looking at Figure 2.16: the fits for the gluon, having more data, looks quite exactly like the ones with cross-validation, while the ones for the c quark, having fewer data points, looks actually more overfitted.

[1] We expect a longer arclength for an overfitted PDF, as it also describe fluctuations.

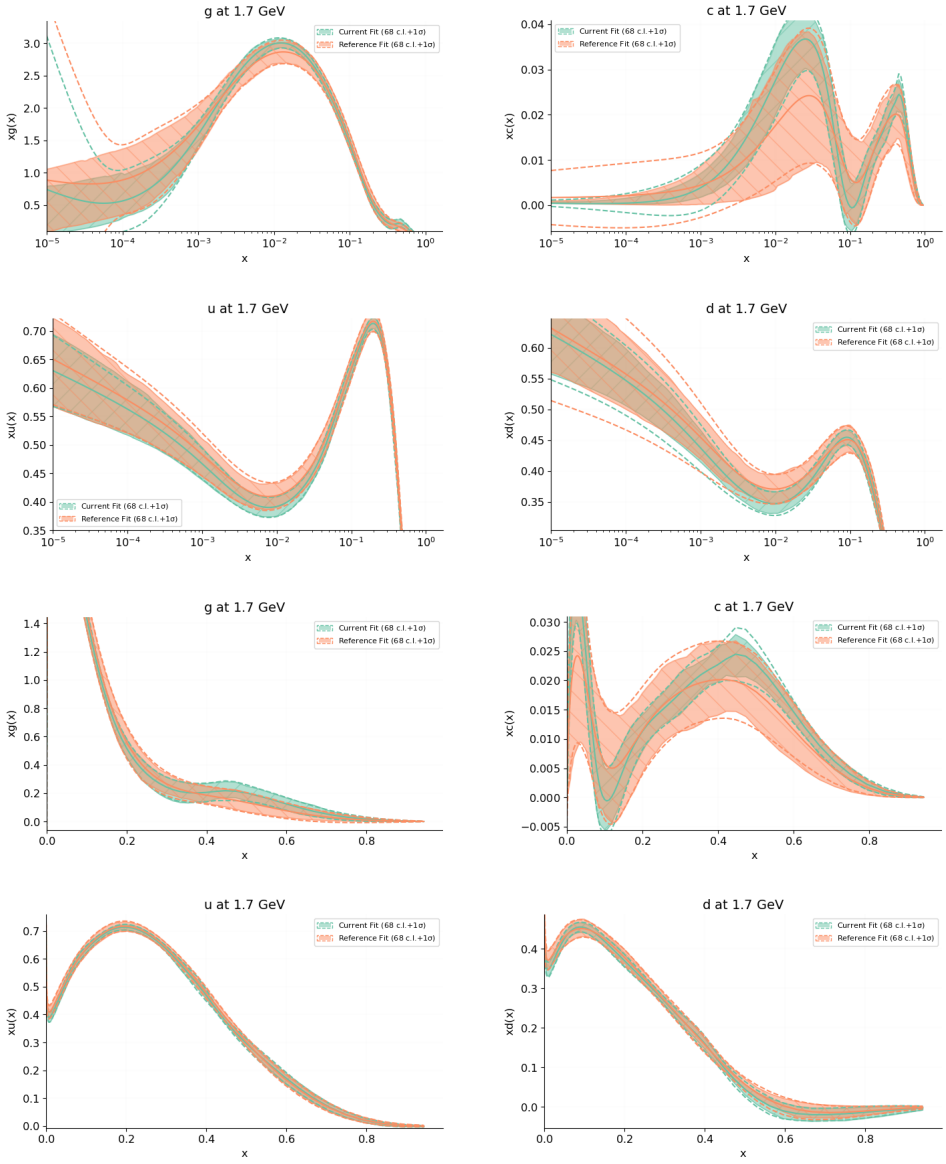

Figure 2.16: PDFs without cross-validation compared with the ones in Section 2.1.1, at the end of the epochs

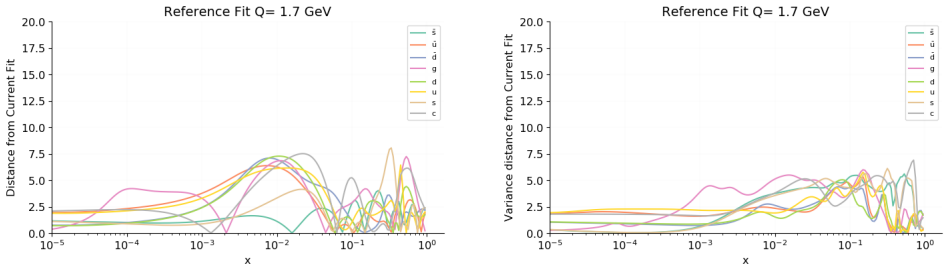

Figure 2.17: Distance between the PDFs without cross-validation and the PDFs at step 200 (left) and variance distance (right)

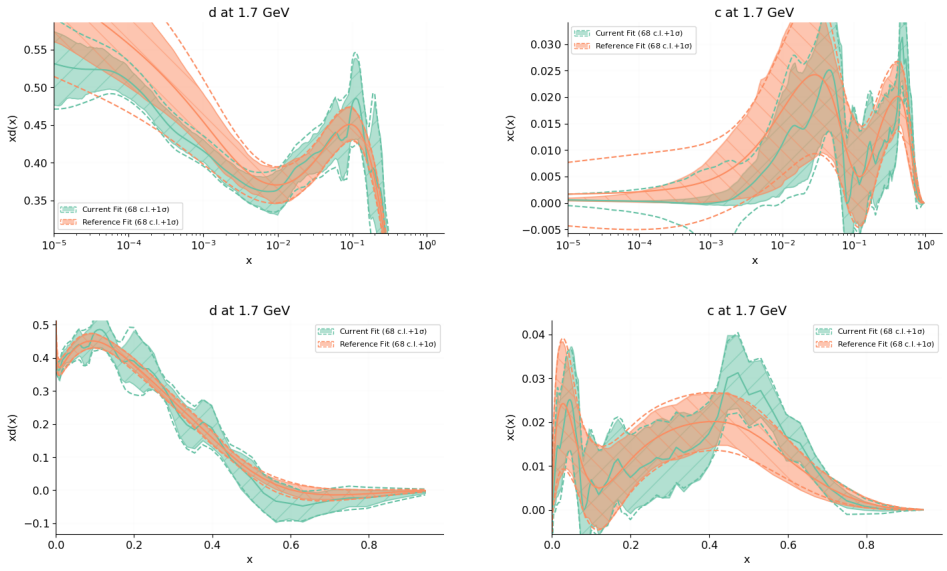

Figure 2.18: Overfitted PDFs compared with the ones in Section 2.1.1, at the end of the epochs

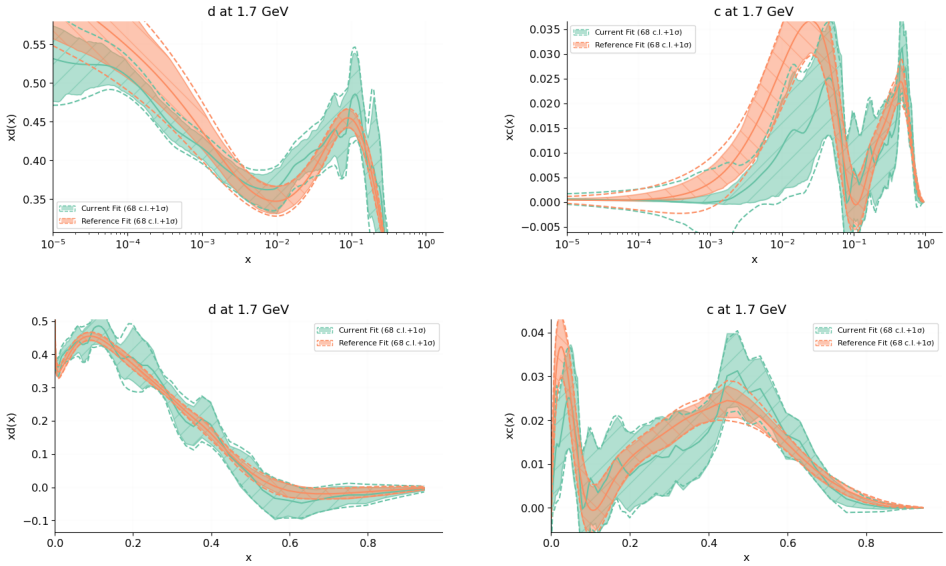

Figure 2.19: Overfitted PDFs compared with the ones without cross-validation, at the end of the epochs

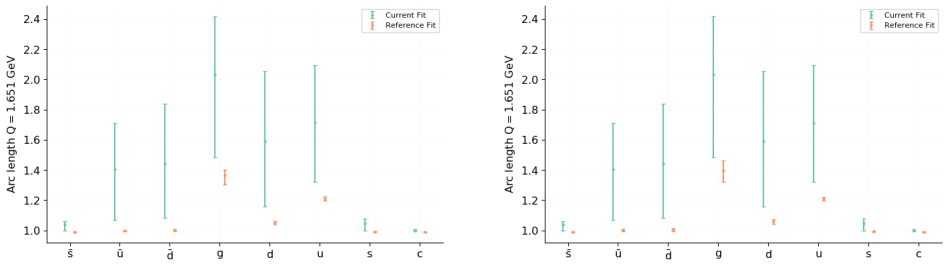

Figure 2.20: Arclengths of overfitted NN compared to the normal one (left) and the one without cross-validation (right)

Therefore, even if cross-validation has a marginal role in preventing overfitting it still has an importan role in the optimisation of the stopping point: thanks to cross-validation it is possible to stop the fit after a suitable number of epochs, for instance it is possible to stop the fit after about 15000 or 20000 epochs instead than after 40000, saving some valuable computation time.

After proving the small importance of cross-validation to avoid overfitting, we try to find what other factors prevent the neural network from overfitting. In Section 2.3 we evaluate the importance of the positivity constrains.

2.3 Positivity

At first, we modify positivity threshold: it represents the minimum value below zero that the fits are allowed to have. In the other fits we used a positivity threshold of 0.01, now we evaluate how modifying this value may change the fit. We used thresholds of 0.1 and of 1, still using no cross-validation. In Figure 2.21 we can see that there is no difference between the two fits because the data and the pseudodata of positivity force the fit to be always over the threshold.

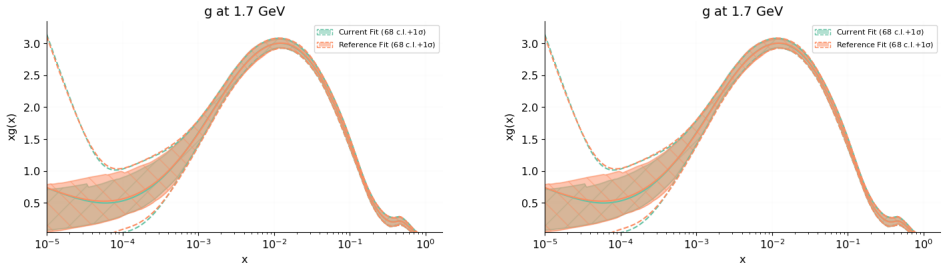

Figure 2.21: PDFs with positivity threshold of 0.1 (left) and 1 (right) compared to threshold of 0.01

After that, we focus on positivity multiplier: it represents the contribution of the pseudodata for χ^2: higher values of positivity multiplier lead to a greater penalty to χ^2 for negative data. Positivity multiplier $\lambda_{multiplier}$ works by increasing the value of the positivity cross-section each 100 epochs. In other words if in step 100 the positivity cross-section is $\sigma_{pos} = x$, then at epoch 1000 it will be $\sigma_{pos} = \lambda^{10}_{multiplier} x$. For the other fits we have used a multiplier of 1.09, now we evaluate the fits with smaller positivity multipliers: 1.05, 1.01, 1.005 and 1.001. In other words, we let the fits be more negative by imposing less restrictive constraint of positivity.

In Figure 2.22 we can see that there is no significant difference by modifying the positivity to only 1.05, to have significative improvements we need to decrease it more. Also all the fits respect the positivity constraint everywhere: the cross-sections predicted by this fits are always positive.

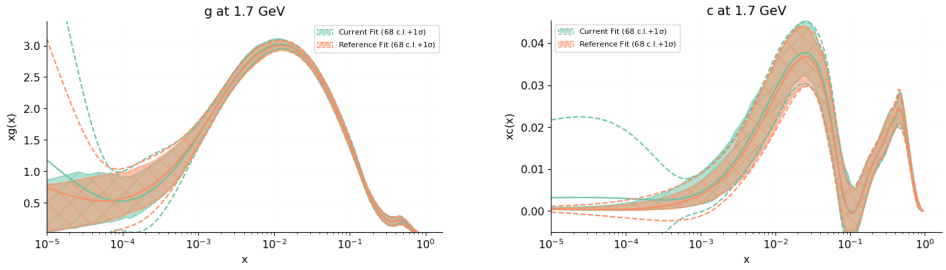

Figure 2.22: PDFs with positivity multiplier 1.05 compared to 1.09

Considering now a positivity threshold of 1.01, we begin to notice some important change in the shape of the fits, as shown in Figure 2.23. Here, the fits start to share some characteristics with the overfitted one of Section 2.2.1. By relaxing the positivity constrain, we now have some fits that converge to a value of small χ^2, so it describes quite well the experimental data, but the solutions are not physical: in fact, in Figure 2.24, we can observe that it predicts negative cross-sections.

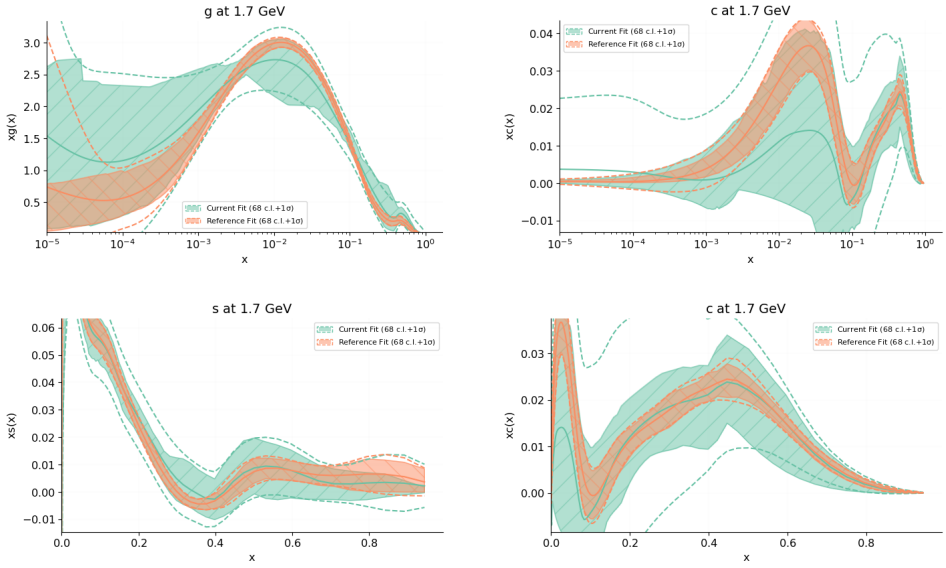

Figure 2.23: PDFs with positivity multiplier 1.01 compared to 1.09

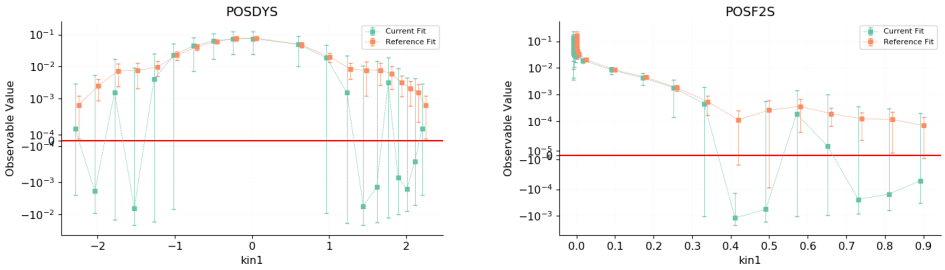

Figure 2.24: Positivity of cross-sections with multiplier of 1.01

If we keep relaxing the positivity multiplier to 1.005 or 1.001 we can notice something different: the results are stopped before the optimal stopping point. This happens because, after a certain point, the χ^2 still decreases, but they don't respect positivity: this way, by having a positivity criteria a family of solutions with a small χ^2 but with unphysical predictions is killed. In Figure 2.25 we can see that the fits actually look stopped before the optimal time and in Figure 2.26 we can see that many cross-sections are negative. While using smaller positivity multipliers, positivity threshold has actually an important role: without relaxing positivity threshold below the initial value of 0.01 all the fits would stop already after the first steps, without giving any significant result.

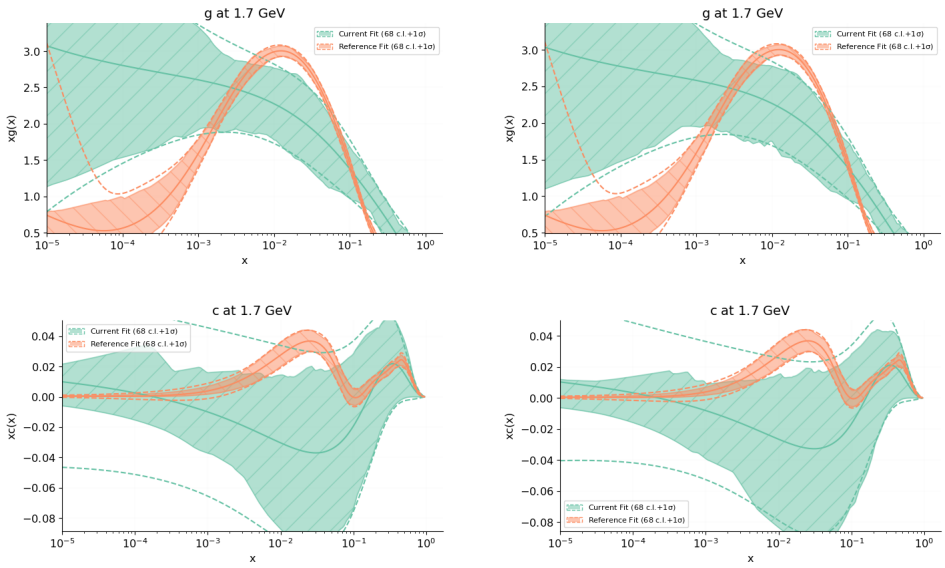

Figure 2.25: PDFs with positivity multiplier 1.005 (left) and 1.001 (right) compared to 1.09

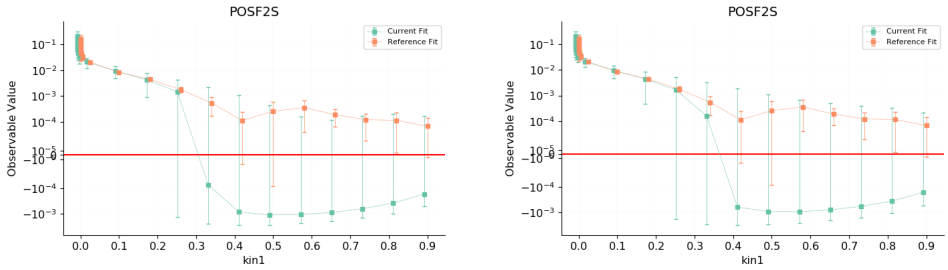

Figure 2.26: Positivity of cross-sections with multiplier of 1.005 (left) and 1.001 (right)

In conclusion, positivity constrain has an important role in eliminating solutions making non-physical predictions, despite their potentially good value of χ^2. Instead, its role in preventing overfitting is still marginal: considering also the marginality of cross-validation, we can conclude that overfitting is prevented mostly by the network used which does't have enough flexibility to overfit.

Conclusion

We can notice that the NNPDF fits converge at the beginning quite quickly, going from $\chi^2 \approx 3$ to a good central value with a $\chi^2 = 1.2$ in about 5000 epochs. After that, the convergence slows down, needing about the same time, or more, to get to the final value of $\chi^2 = 1.1$. Furthermore, not all the replicas reach the stopping point at the same time. By looking at the different stopping points, we can conclude that a suitable stopping point could be at about 15000 or 20000 epochs, where most of the replicas have reached their final value, allowing to save some computational time with respect to the 40000 epochs used in this work. Furthermore, during the evolution, the 1-σ contour becomes smaller with the evolution of the network. In particular, at the end we have an error way smaller than the error at the beginning. This is due to the fact that at the beginning the neural networks is describing data randomly, so it fluctuates more, leading to a bigger contour.

Removing cross-validation, we observed that its importance is negligible. This happens for two main reasons: the first one is that the neural network is trained on a huge amount of data. This way, the random fluctuations are flattened out, so that the neural network describes an average of these fluctuations, without overfitting on the single points. The second reason is that the neural network is too small to overfit; in fact, using a bigger network it is possible to have overfitting. Cross-validation still has an important role in determining the optimal stopping point.

After concluding the negligible role of cross-validation for avoiding overfitting, we evaluated if the positivity constrain plays a role in overfitting, concluding that it does, but only marginally. In fact, by relaxing the positivity multiplier, the fits started looking a little overfitted, but not as much as the results using the bigger network. It is a marginal role because by relaxing more this constrain, the fits are stopped before the optimal stopping-point, resulting in under-learning. This way, a family of solutions with small χ^2 but making non-physical predictions, like negative cross-sections, is killed.

Bibliography

[1] Forte, S., & Watt, G. (2013). *Progress in the determination of the partonic structure of the proton*. Annual Review of Nuclear and Particle Science, 63, arXiv:1301.6754.

[2] Roberts, R. G. (1990). *The structure of the proton*, Cambridge Univ.

[3] Giele, W. T., Keller, S. A., & Kosower, D. A. (2001). *Parton distribution function uncertainties*. arXiv preprint arXiv:hep-ph/0104052.

[4] Cowan, G. (1998). *Statistical data analysis*. Oxford university press.

[5] NNPDF Collaboration, Del Debbio, L., Forte, S., Latorre, J. I., Piccione, A., Rojo, J. (2007). *Neural network determination of parton distributions: the nonsinglet case*. Journal of High Energy Physics, 2007(03), 039, arXiv:hep-ph/0701127.

[6] NNPDF Collaboration, Ball, R. D., Bertone, V., Cerutti, F., Del Debbio, L., Forte, S., Guffanti, A., ... (2012). *Unbiased global determination of parton distributions and their uncertainties at NNLO and at LO*. Nuclear Physics B, 855(2), 153-221, arXiv:1107.2652.

[7] Forte, S., Garrido, L., Latorre, J. I., & Piccione, A. (2002). *Neural network parametrization of deep-inelastic structure functions*. Journal of High Energy Physics, 2002(05), 062, arXiv:hep-ph/0204232.

[8] NNPDF Collaboration, Ball, R. D., Bertone, V., Carrazza, S., Deans, C. S., Del Debbio, L., Forte, S., ... (2015). *Parton distributions for the LHC Run II*. Journal of High Energy Physics, 2015(4), 40, arXiv:1410.8849.

[9] NNPDF Collaboration, Ball, R. D., Bertone, V., Cerutti, F., Del Debbio, L., Forte, S., Guffanti, A., ... (2011). *Impact of heavy quark masses on parton distributions and LHC phenomenology*. Nuclear Physics B, 849(2), 296-363, arXiv:1101.1300.

[10] Altarelli, G., Forte, S., & Ridolfi, G. (1998). *On positivity of parton distributions*. Nuclear physics B, 534(1-2), 277-296, arXiv:hep-ph/9806345.

[11] NNPDF Collaboration, Ball, R. D., Del Debbio, L., Forte, S., Guffanti, A., Latorre, J. I., Rojo, J., ... (2010). *A first unbiased global NLO determination of parton distributions and their uncertainties*. Nuclear Physics B, 838(1-2), 136-206. arXiv:1002.4407.

[12] NNPDF Collaboration, Ball, R. D., Bertone, V., Carrazza, S., Del Debbio, L., Forte, S., Groth-Merrild, P., ... (2017). *Parton distributions from high-precision collider data*. The European Physical Journal C, 77(10), 663, arXiv:1706.00428.

[13] Carrazza, S., Cruz-Martinez, J. (2019). *A new generation of parton distribution functions with deep learning models*. Still unpublished.