NNPDF4.0

Neural networks techniques for parton distribution functions evaluation

Andrea Barontini on behalf of the NNPDF collaboration

Alpaca: modern algorithms in machine learning and data analysis: from medical physics to research with accelerators and in underground laboratories

20/11/2023

Based on: hep-ph:1907.05075,2109.02653,2208.08372,1906.10698,1509.00209

Machine Learning • PDFs • QCD

Outline

- The Physics
- The NNPDF framework
- Results and outlook

Outline

The Physics

The NNPDF framework

Results and outlook

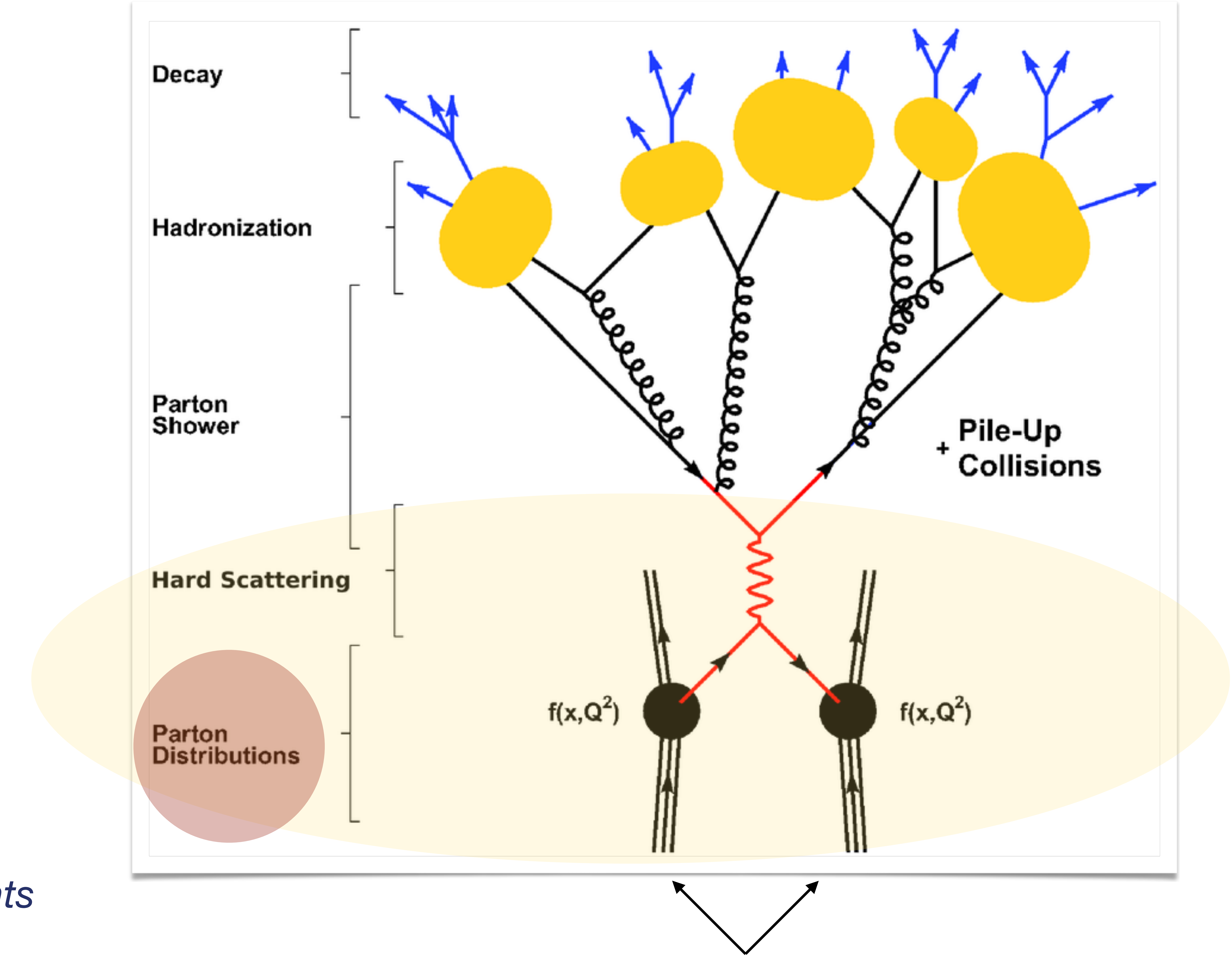
Describing a collision

The theoretical description of a **collision** involves several **QCD** (Quantum ChromoDynamics) ingredients

We are going to focus on

And, in particular, on Parton Distribution Functions **(PDFs)**

Describe the hadronic initial state in terms of their *partonic components*



Initial state = hadrons (protons, neutrons ,...)

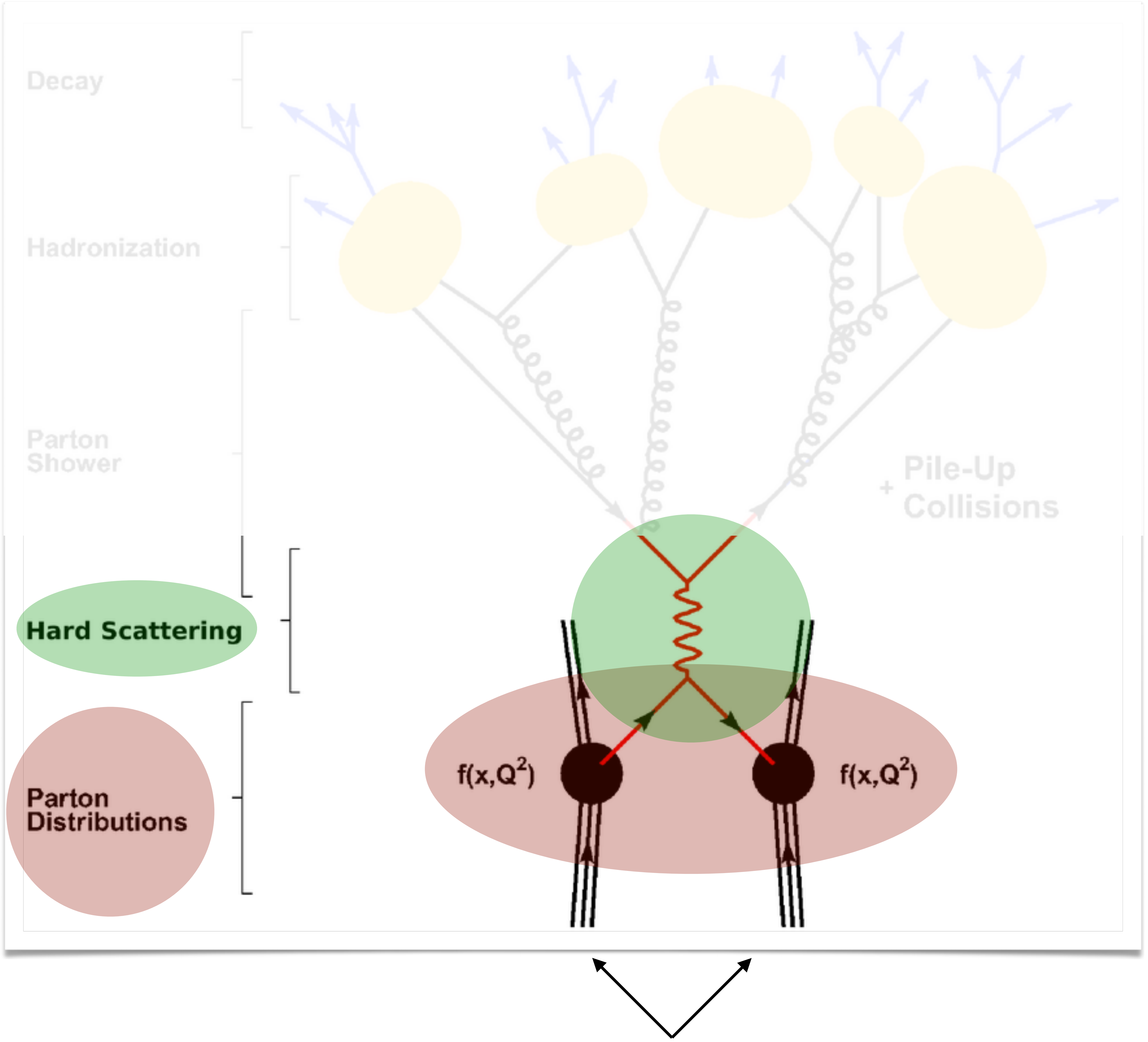
Factorization: divide and conquer

Thanks to **Factorization theorem**

PDFs

$$\sigma(x,Q^2) = \hat{\sigma}_{ij} \otimes f_i \otimes f_j = \int dz_1 dz_2 \hat{\sigma}(z_1,z_2,Q^2) f_i\left(\frac{x}{z_1},Q^2\right) f_j\left(\frac{x}{z_2},Q^2\right)$$

Partonic (hard) cross sections



- $\sigma(x,Q^2)$ is our **observable**
- Q^2 is the energy scale of the process
- $\hat{\sigma}(z_1,z_2,Q^2)$ can be computed in **perturbation theory**
- $f_{i/j}(x,Q^2)$ **cannot** be computed in perturbation theory

  (and they are **universal**)

BUT WHY??

Initial state = hadrons (protons, neutrons ,...)

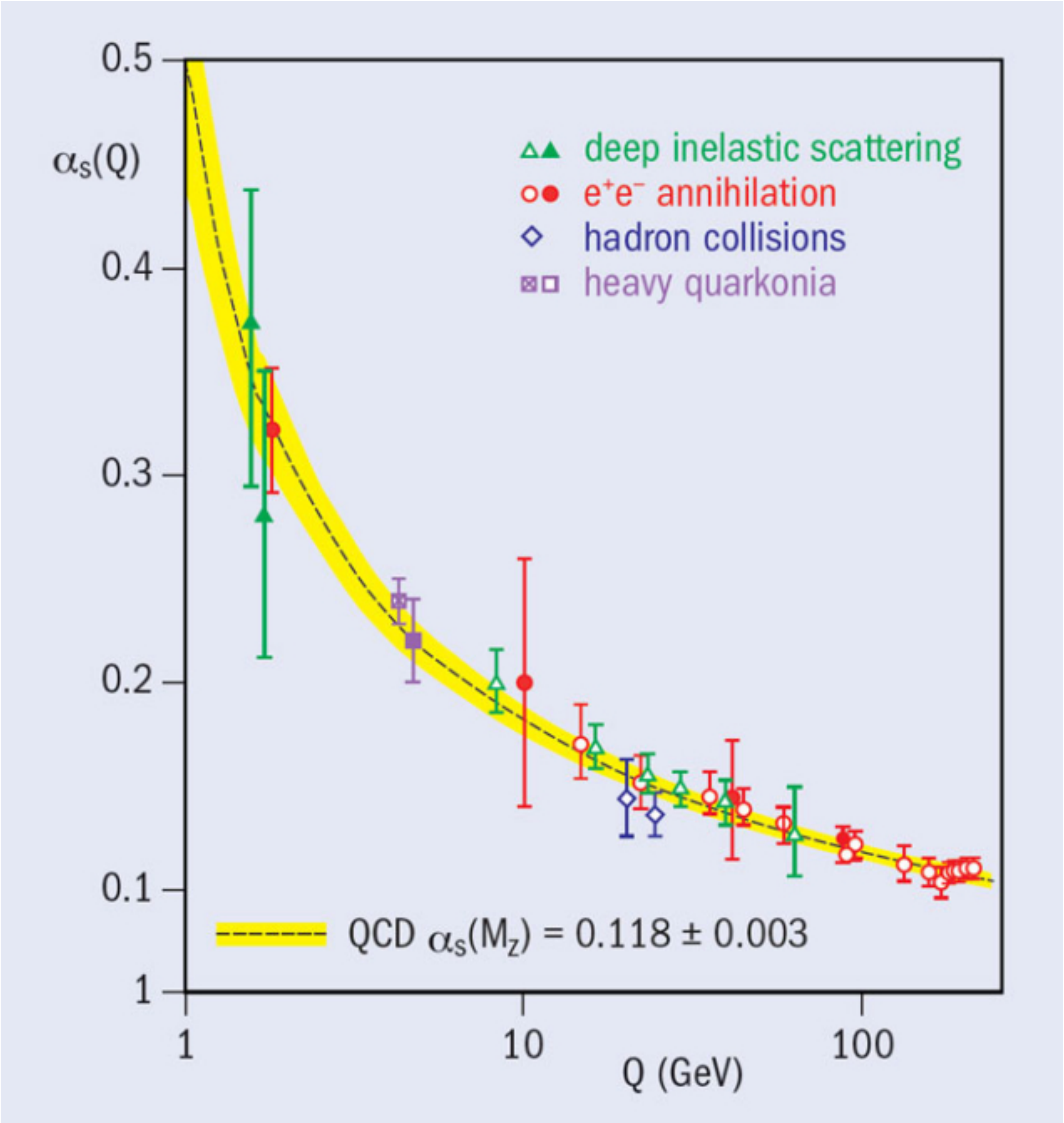
Asymptotic freedom

In **QCD** we are usually expand quantities in terms of the **strong coupling** $\alpha_s(Q^2)$ → (Notable counterexample is lattice QCD)

$$\hat{\sigma}^{NLO}(z_1, z_2, Q^2) = \hat{\sigma}^{(0)}(z_1, z_2, Q^2) + \alpha_s(Q^2)\hat{\sigma}^{(1)}(z_1, z_2, Q^2) + \mathcal{O}(\alpha_s^2)$$

$(NLO =$ Next-to-leading order)

But $\alpha_s(Q^2)$ is a decreasing function of the energy scale



Physik Journal 3 12 31

perturbative QCD
(pQCD)
from ~ 1 GeV

Partonic cross sections

Non perturbative QCD
below ~ 1 GeV

PDFs
(Mass of the proton ~ 0.938 GeV)

How can we extract them?

PDF extraction

Let’s look at the **Factorization theorem** from another prospective

unknown

$$\sigma(x, Q^2) = \hat{\sigma}_{ij} \otimes f_i \otimes f_j = \int dz_1 dz_2 \hat{\sigma}(z_1, z_2, Q^2) f_i\left(\frac{x}{z_1}, Q^2\right) f_j\left(\frac{x}{z_2}, Q^2\right)$$

Inverse problem

Measured in experiments

computed in perturbation theory

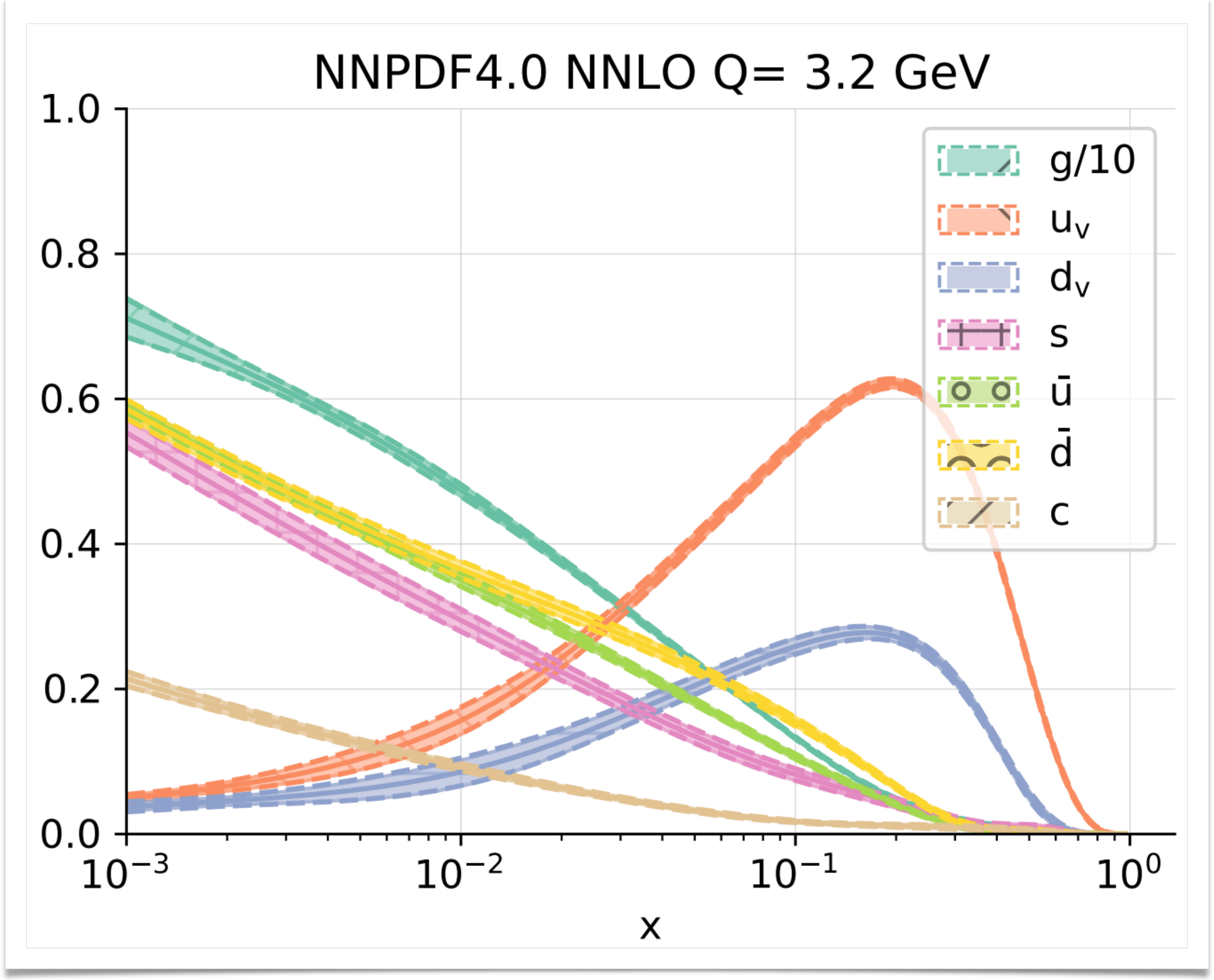
Also, **DGLAP equations** allow us to compute the PDFs at all scale Q^2, once known at a certain scale Q_0^2

$$f_i(Q^2) = E_{ij}(Q^2 \leftarrow Q_0^2) f_j(Q_0^2)$$

PDFs are then just a set of **unknown functions**

$$f_i : [0,1] -> \mathbb{R}$$

$f_i(x) \sim$ probability of extracting parton i from the proton with momentum fraction x



Outline

The Physics

The NNPDF framework

Results and outlook

Inverse problems

Number of datapoints is **finite**[1] while function space is **infinite-dimensional**

Fitting PDFs is always an **under-determined** problem

ASSUMPTIONS

Fixed parametrization

- Reduce the number of parameters
- Assumptions = choice of the parameters to be fitted

Neural Network

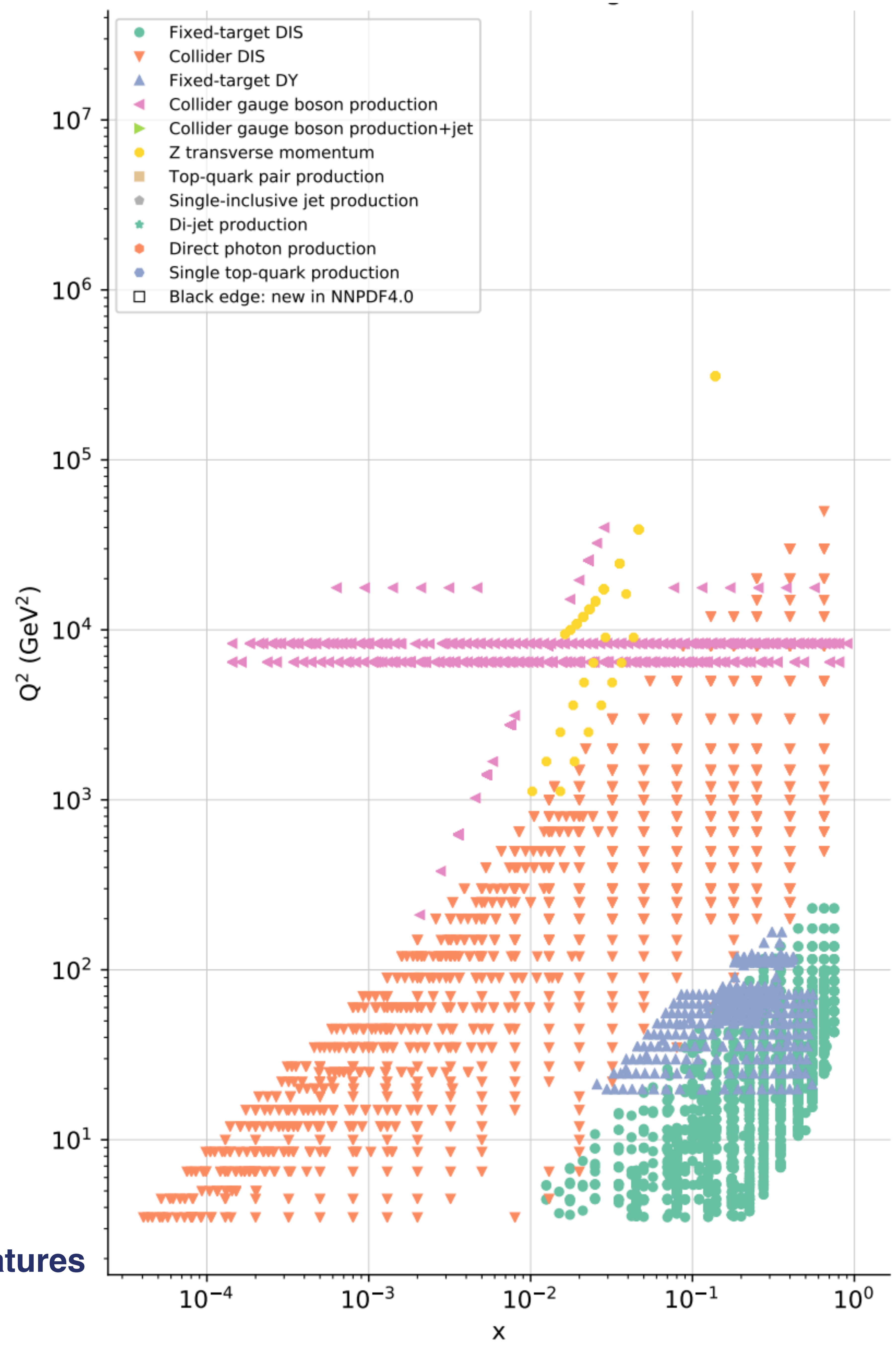
NNPDF choice

- Applies a **regularization**
- Assumptions = encoded in the network (and not only…)

Which is better?

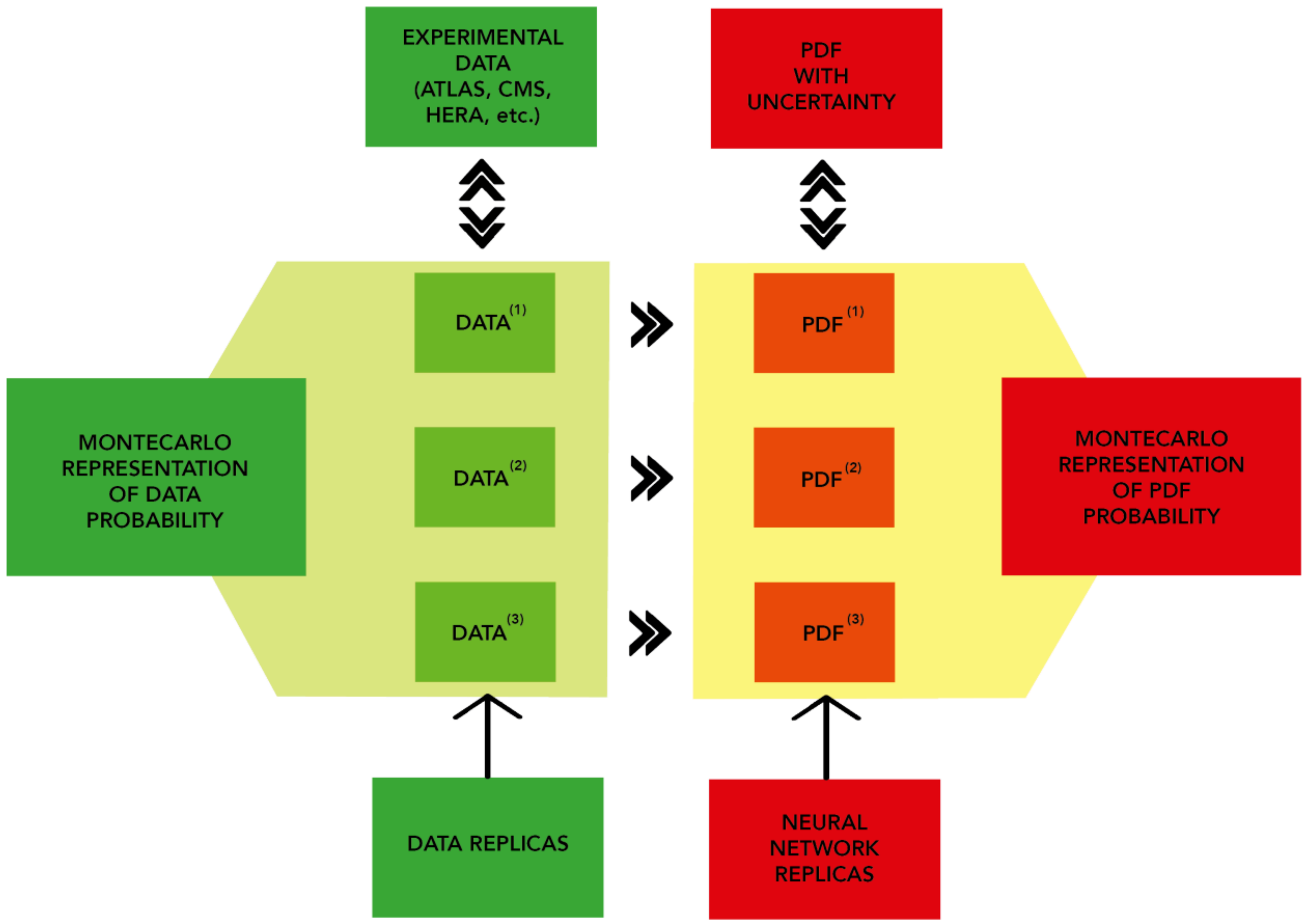
- Needs theoretical insight on PDFs shape
- Can be biased by human prejudice

- Needs theoretical insight on more **abstract features**
- Human prejudice effect can be minimized



[1] ~4600 datapoints in NNPDF4.0

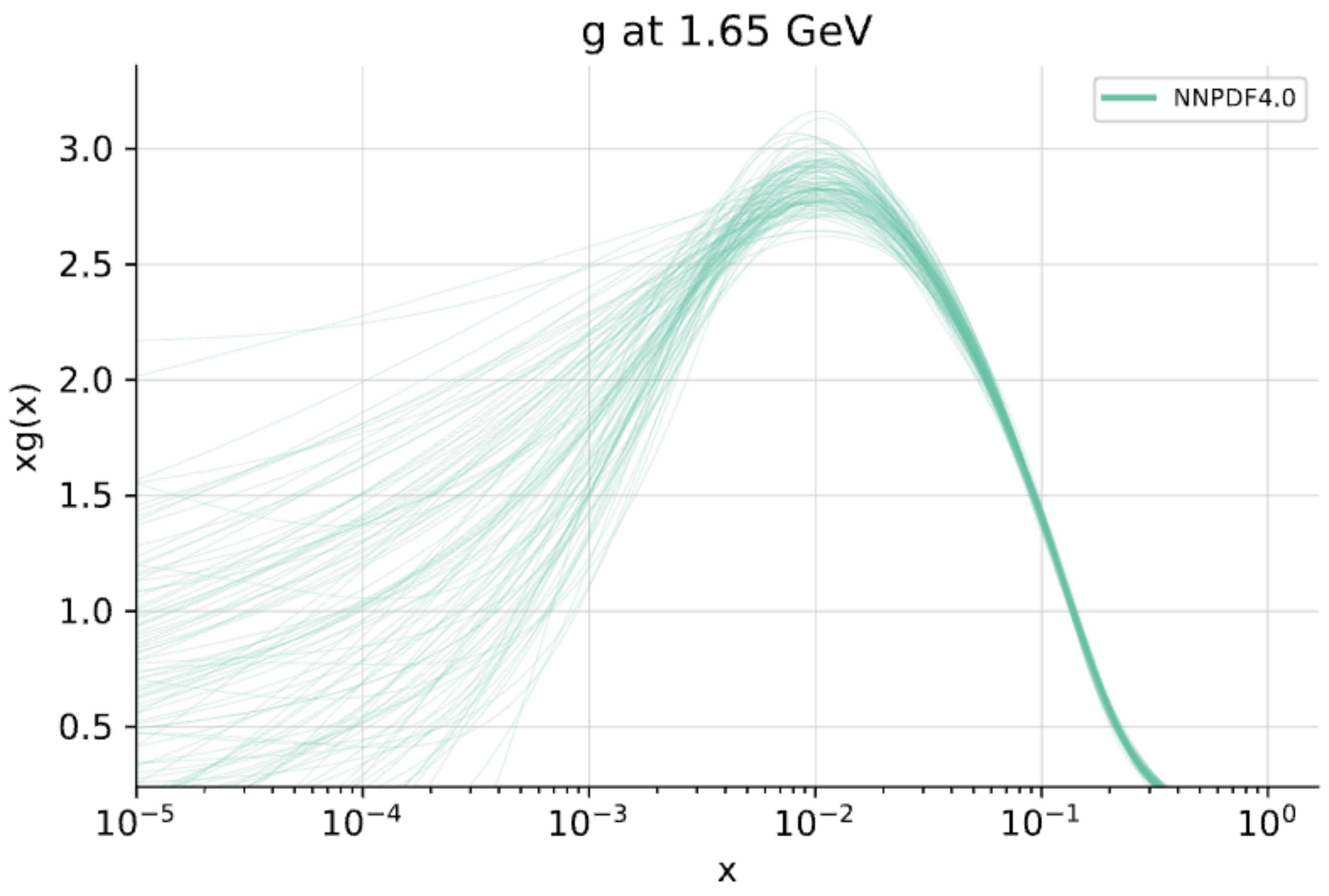
Propagating uncertainties: data to PDFs

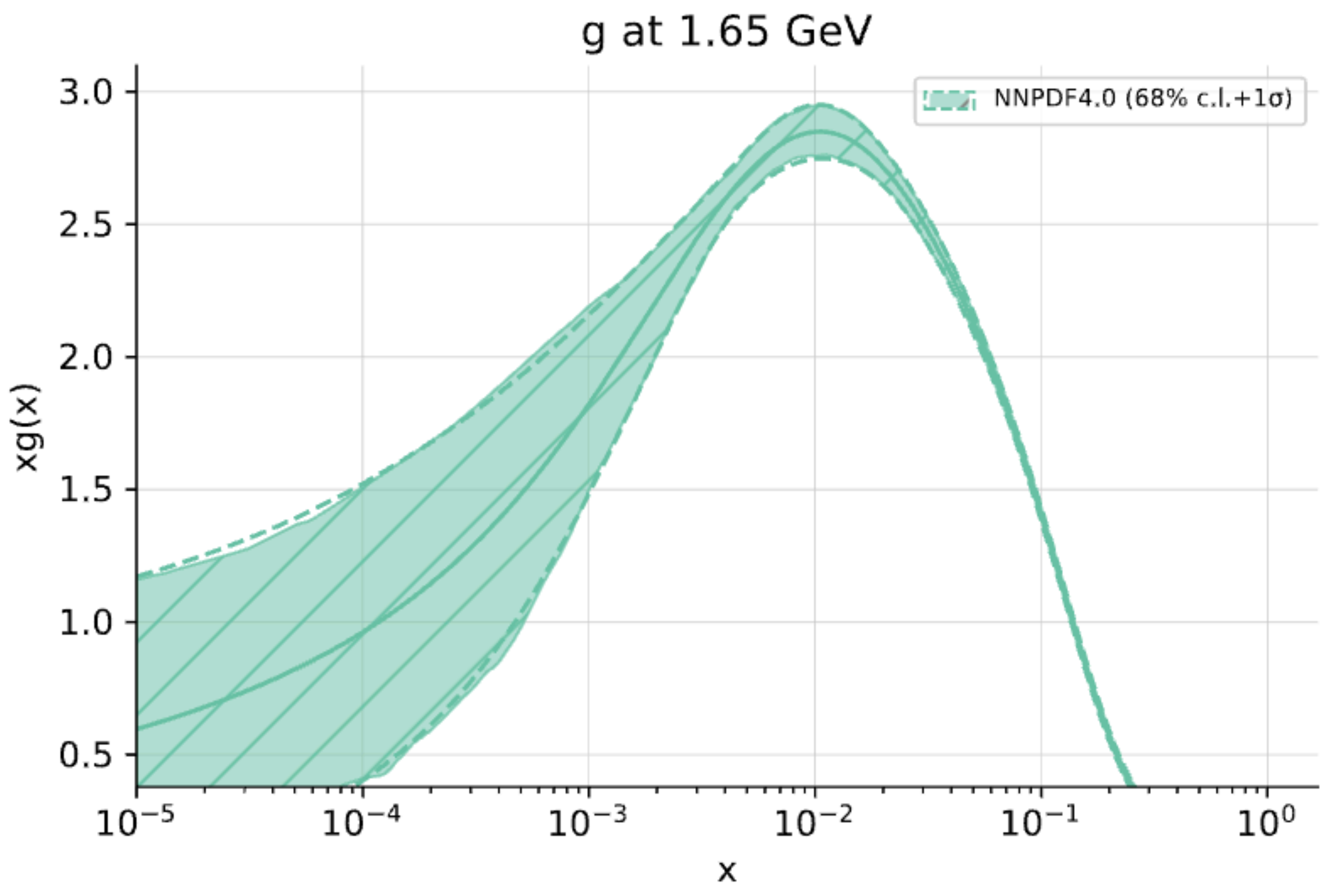


NNPDF adopt a **Monte Carlo** approach

1. Start with the original dataset D and its **covariance matrix** C
2. Generate N_{rep} **pseudodata** D_i according to C
3. Fit a **Neural Network NN$_i$** to each of the pseudodata replica
4. Deliver the full set of replicas

PDFs uncertainties are given by the distribution of the Monte Carlo set





NB: Another possibility is the Hessian approach. The two methods can be converted one in the other (hep-ph:1505.06736)

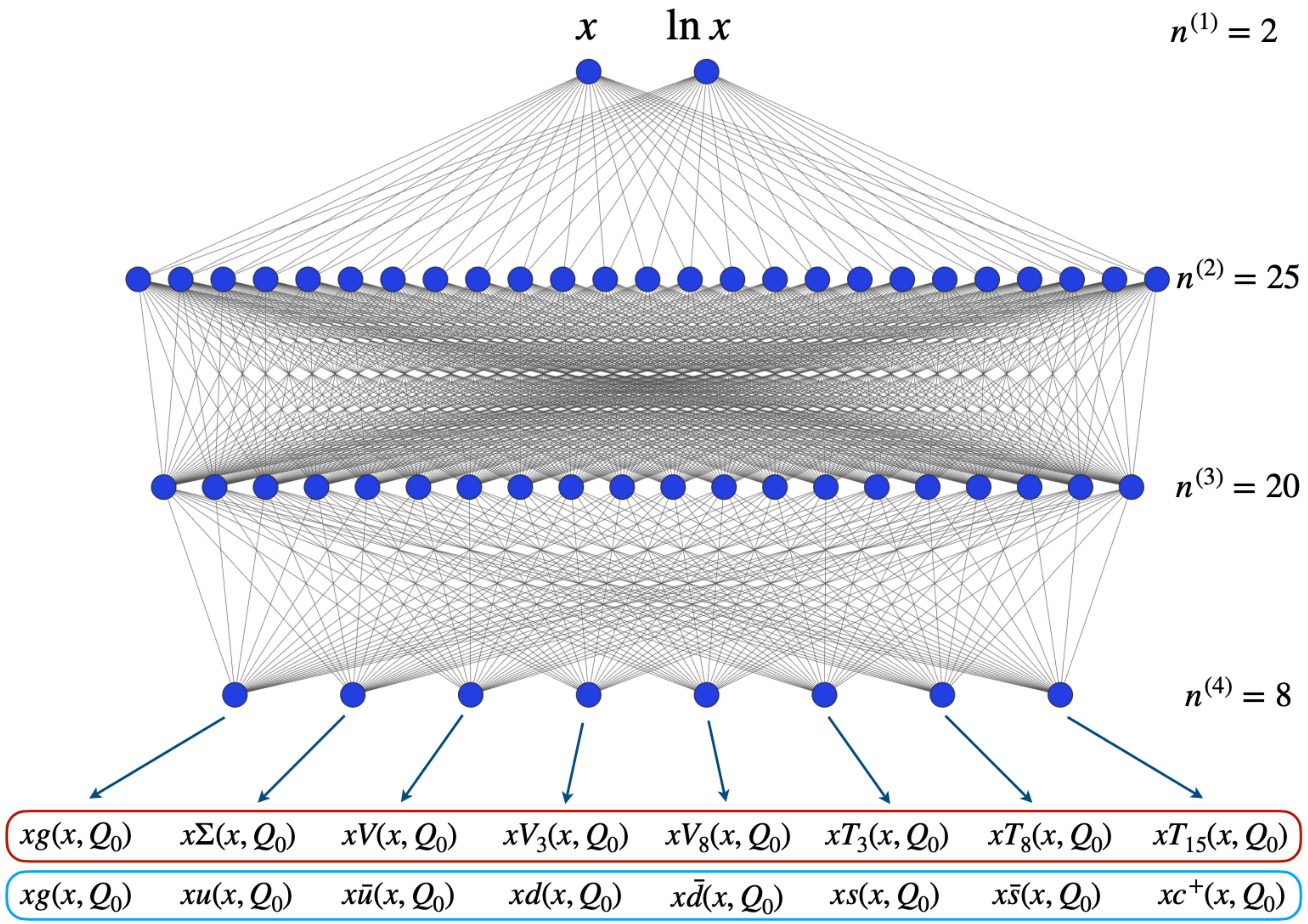
The Neural Network

Architecture: 2-25-20-8
Activation functions: hyperbolic; linear for the last layer
Preprocessing: $A_k x^{-\alpha_k}(1-x)^{\beta_k}$
Optimizer: Adadelta
Physics assumptions:

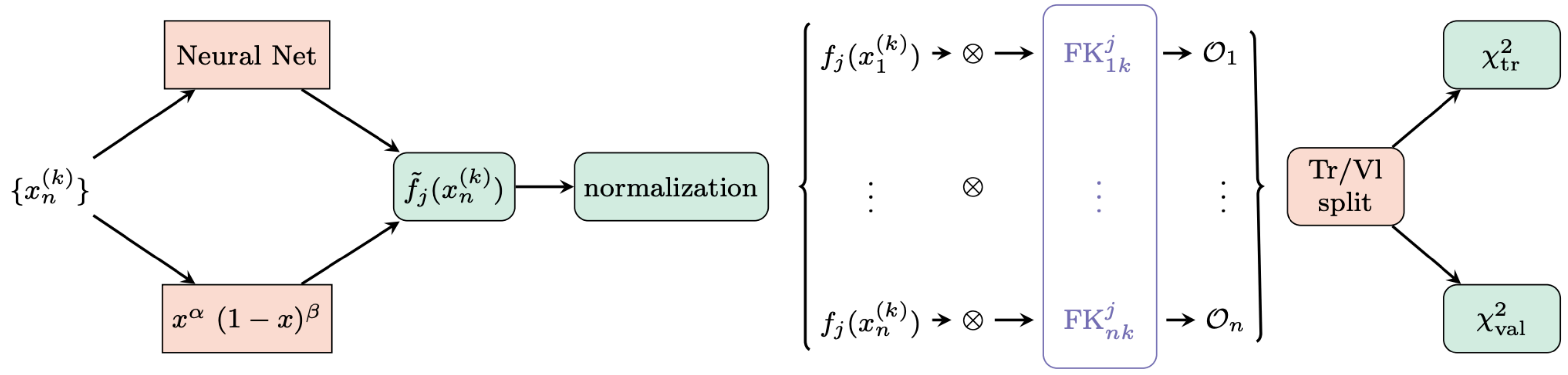
→ **Sum Rules** $\int_0^1 dx V(x,Q) = 3$

→ **PDF positivity**

→ **Integrability**



Neural Network: universal interpolator



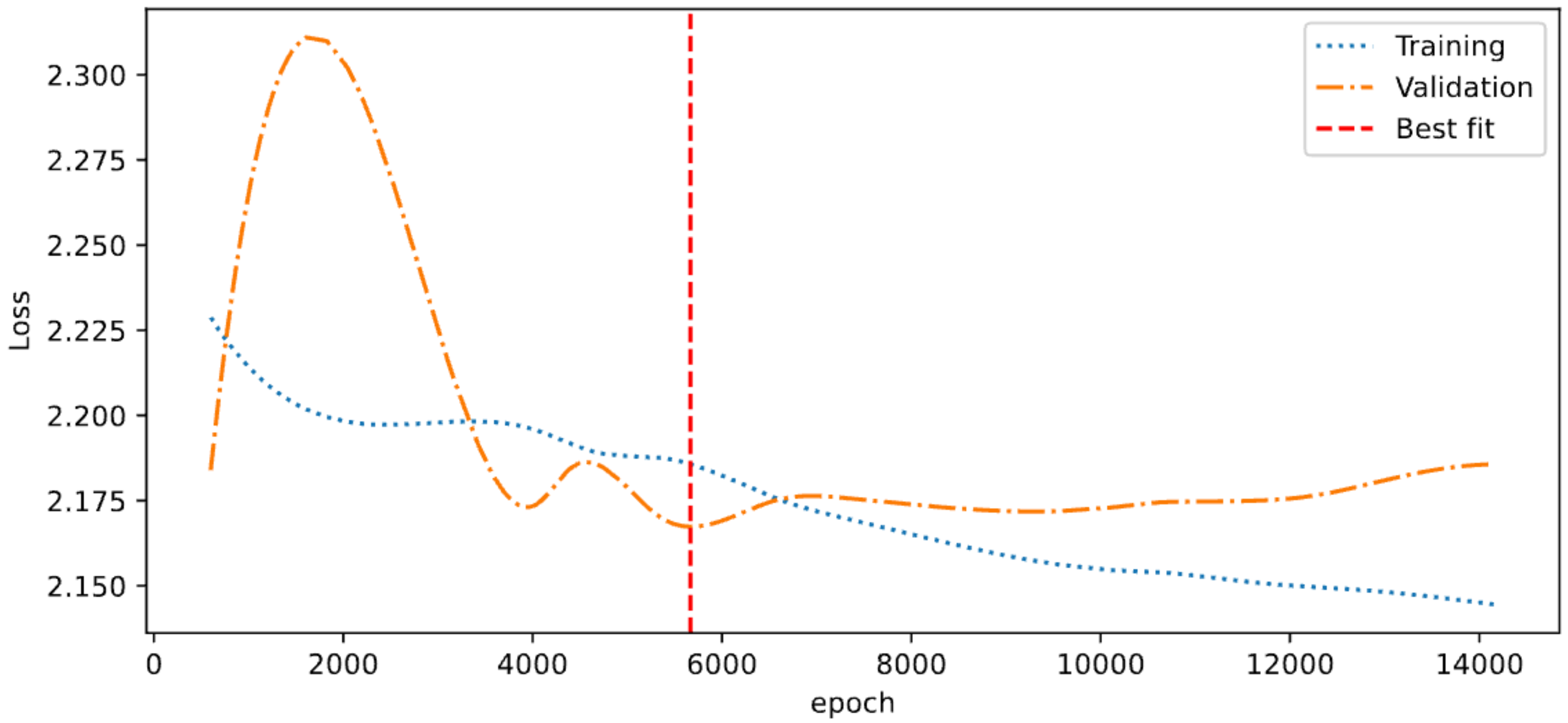
The Neural Network: training

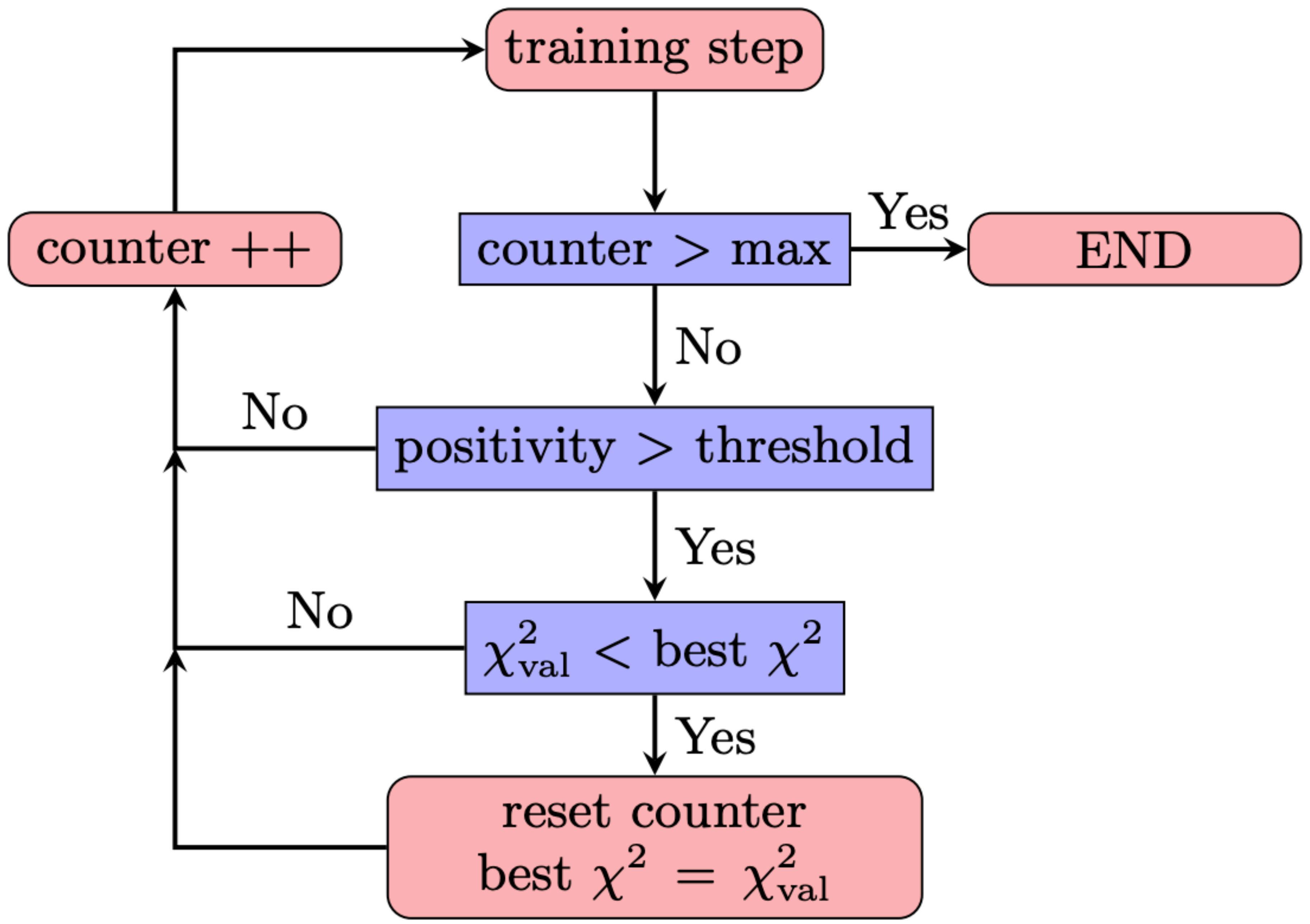
Avoid **overfitting** (fitting the noise)

Cross-validation

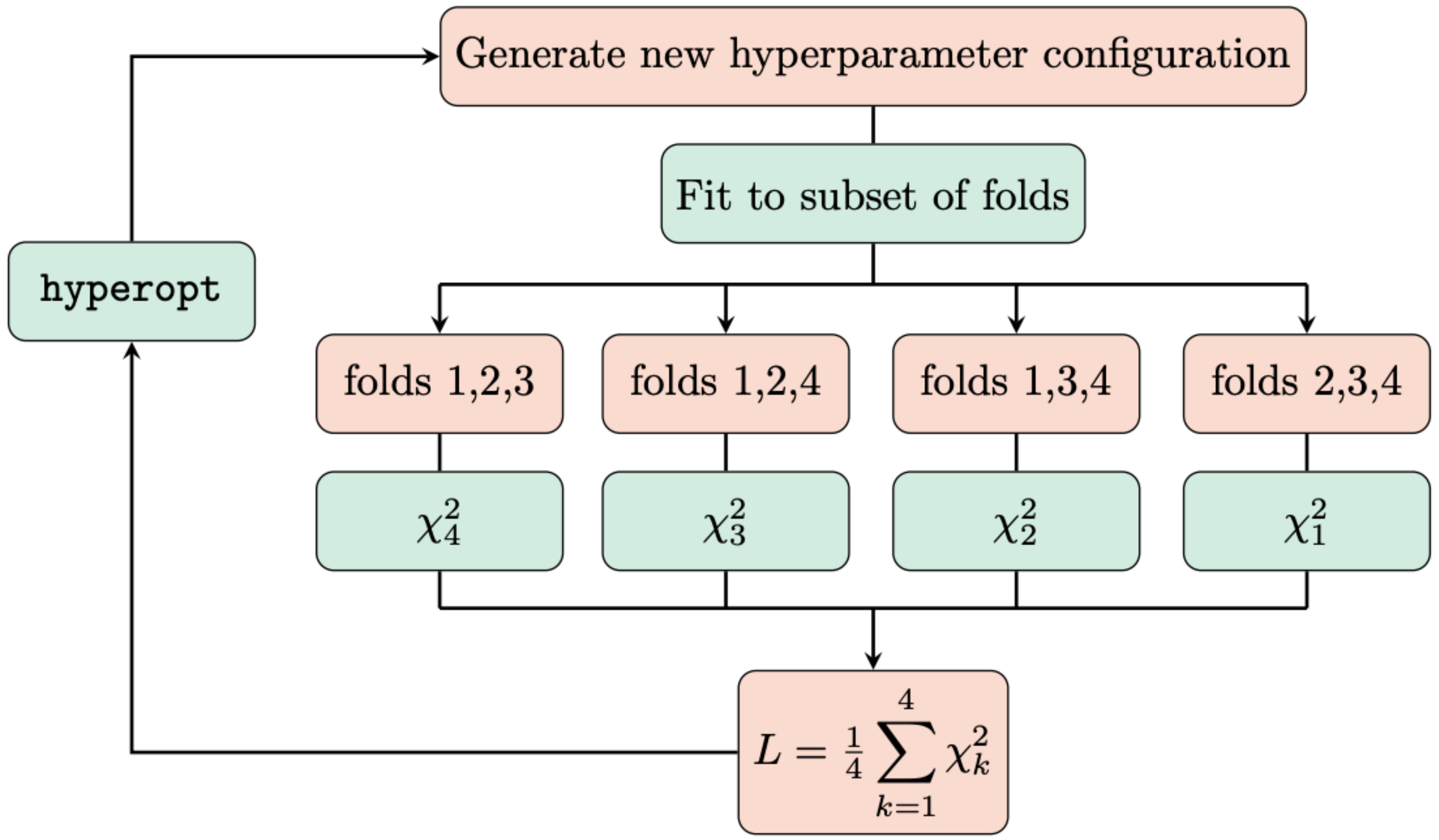
Stopping

1. Divide data into **training** and **validation**
2. Minimize training χ^2
3. Stop if validation χ^2 no longer improves





Automated model selection



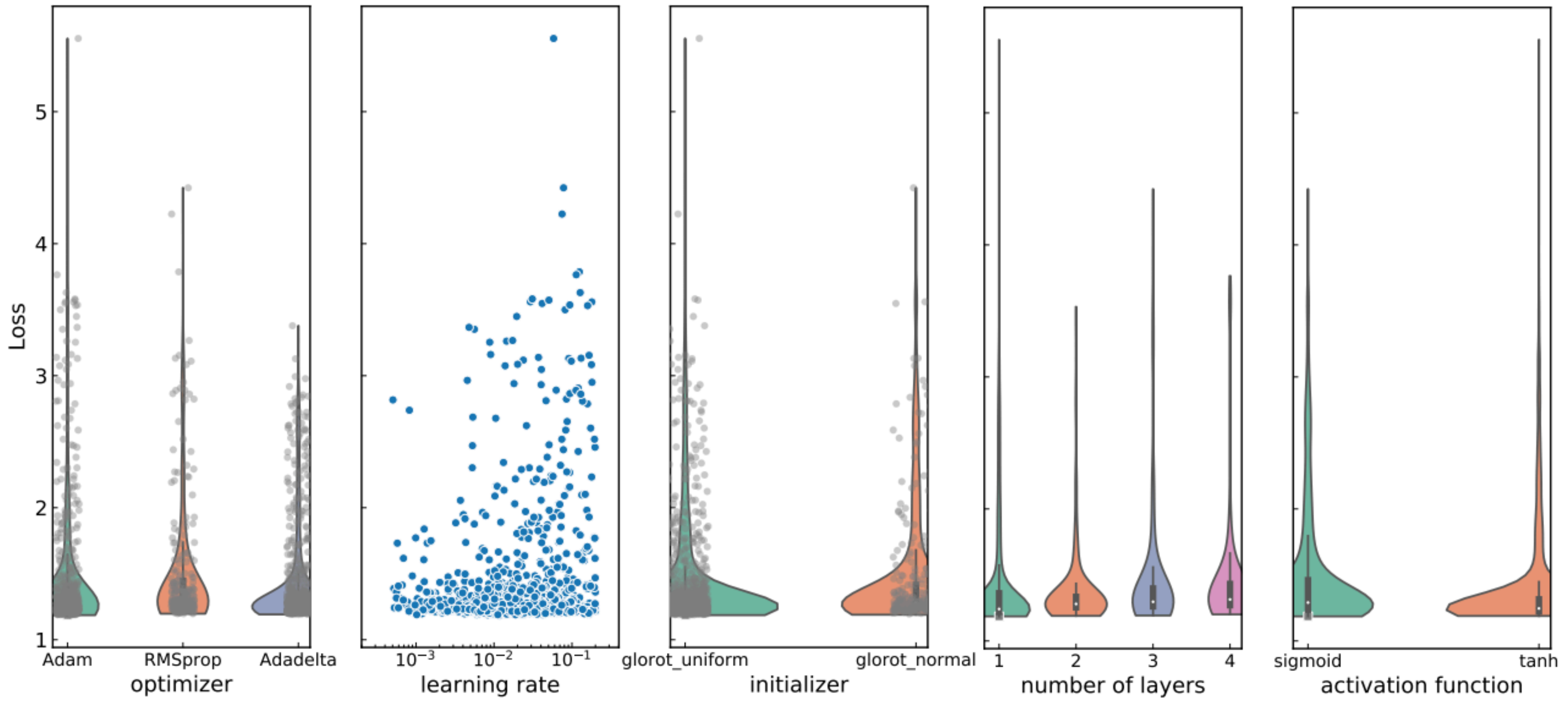
Minimize sources of **bias** in the PDFs:

- Functional form → Neural Network
- Model parameters → **Hyperoptimization**

Idea is to scan over a large enough hyperparameter space and select the best set

Best → best χ^2 on a **test dataset** (never seen by the NN)

NB: Still requires some human input (more on this later)



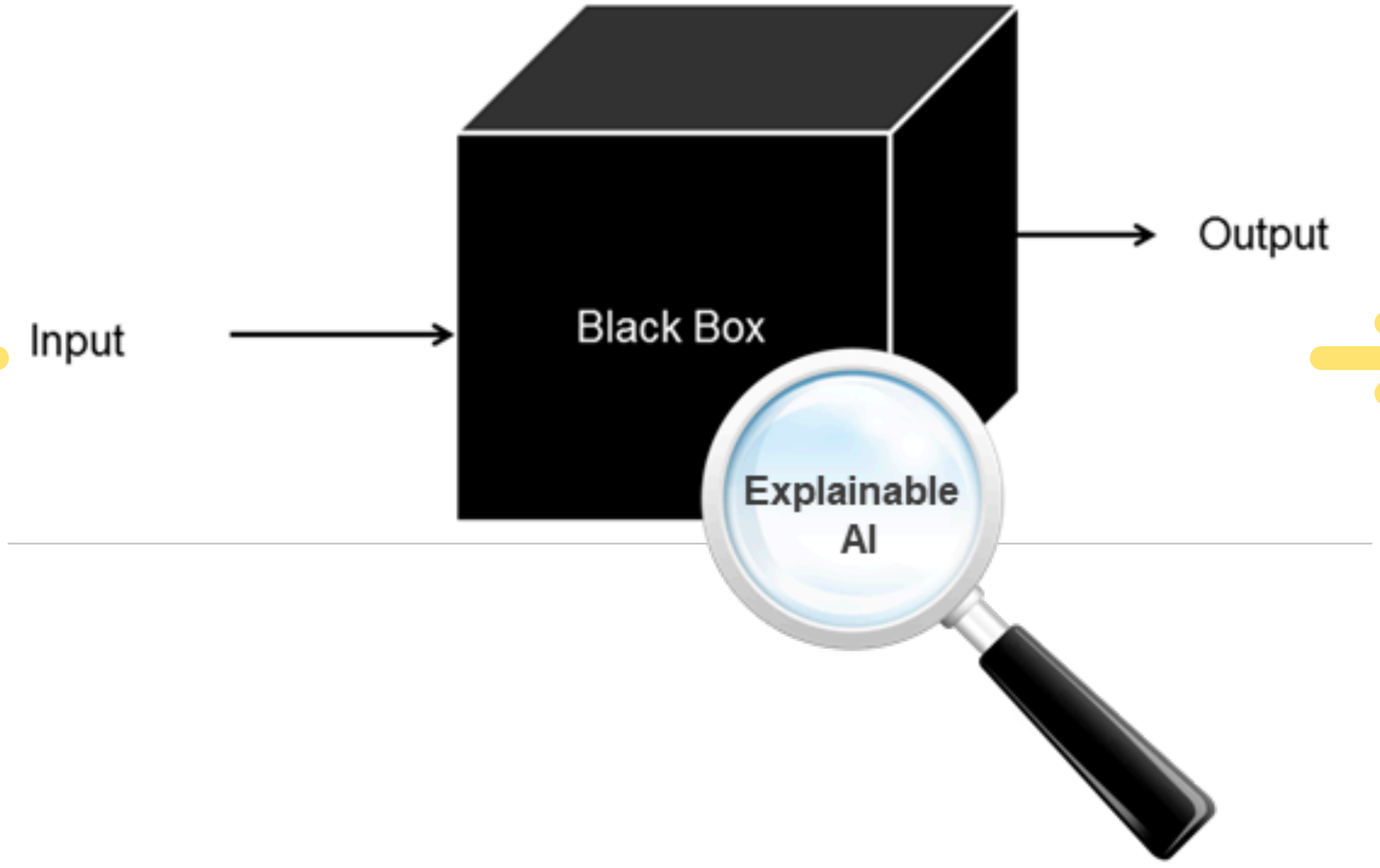
Can we trust our results?

Downside of Neural Networks:
we lack a **full analytical insight** on the process

NN is often considered to be a **black box**

Tests a priori (WIP)

- Test internal features of the NN
- "Analytical" approach



Tests a posteriori

- Test properties of the results
- Empirical approach

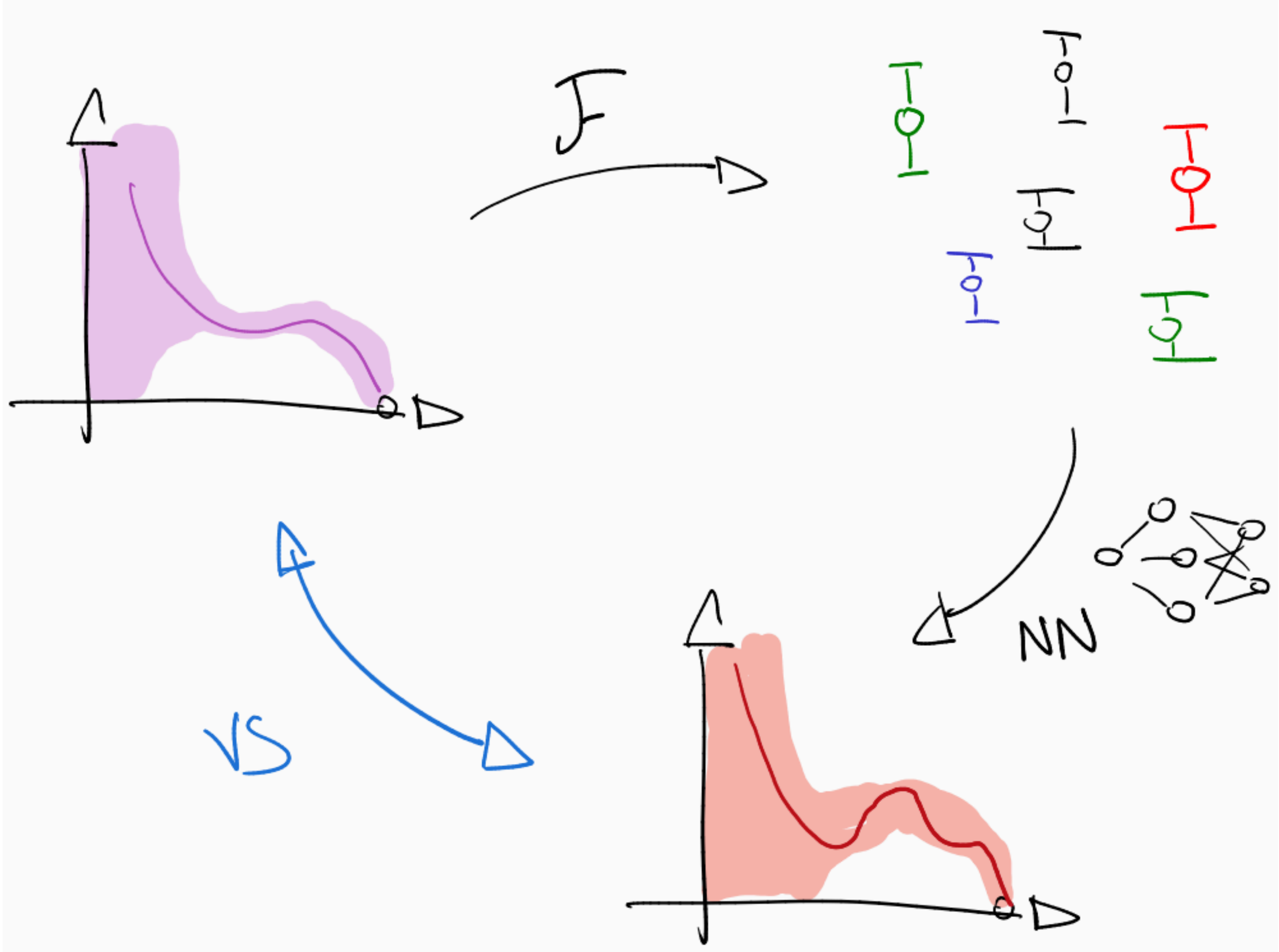
Focus on these!

Closure and future tests

Closure test

Test the algorithm in a controlled environment where the **"truth" is known**

1. Choose a PDF as underlying truth
2. Generate central fake data (**LEVEL 0**)
3. Generate smeared fake data with the experimental covariance matrix (**LEVEL 1**)
4. Generate and fit pseudodata replica (**LEVEL 2**)
5. Compare the results with known distribution

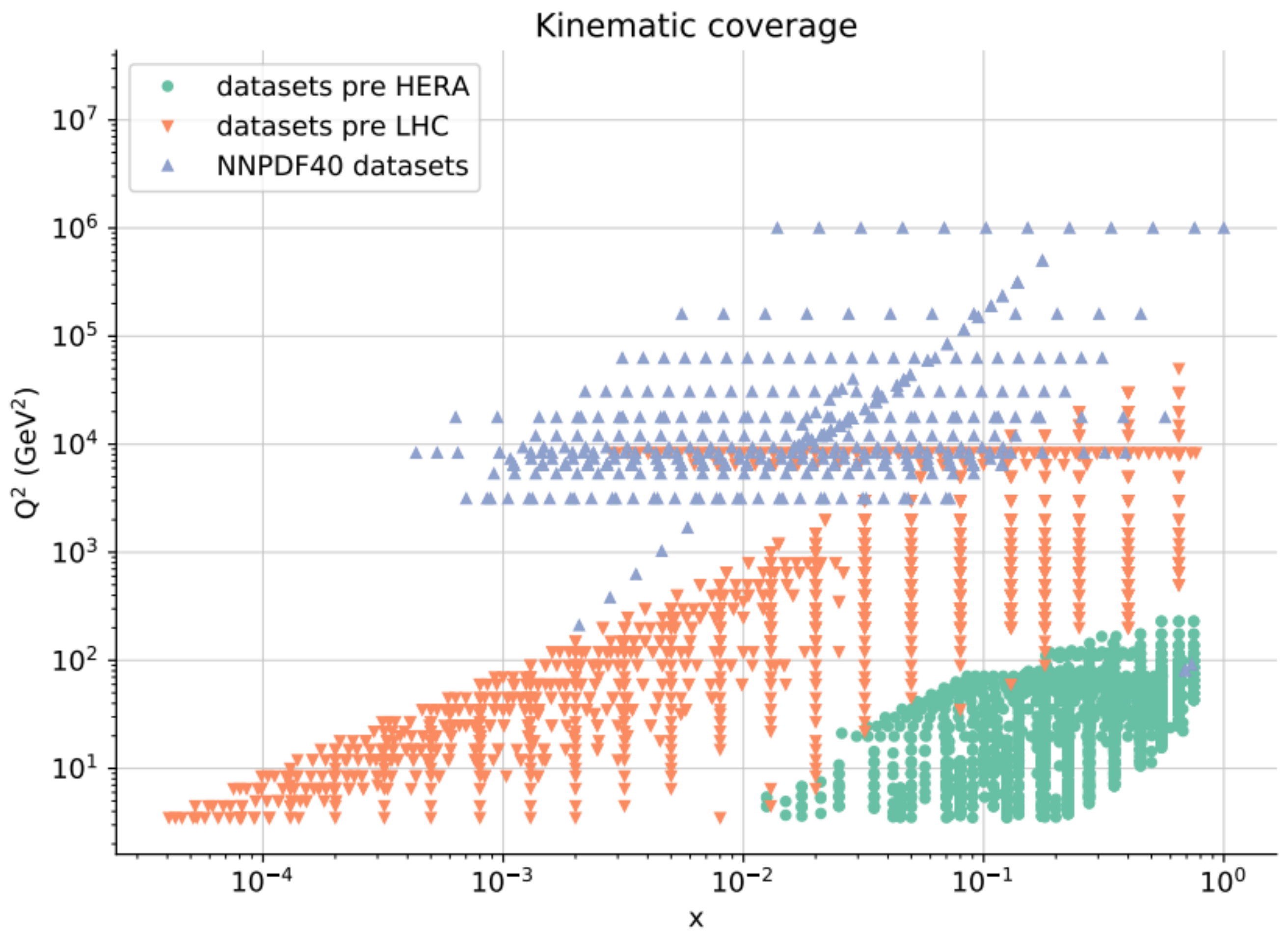


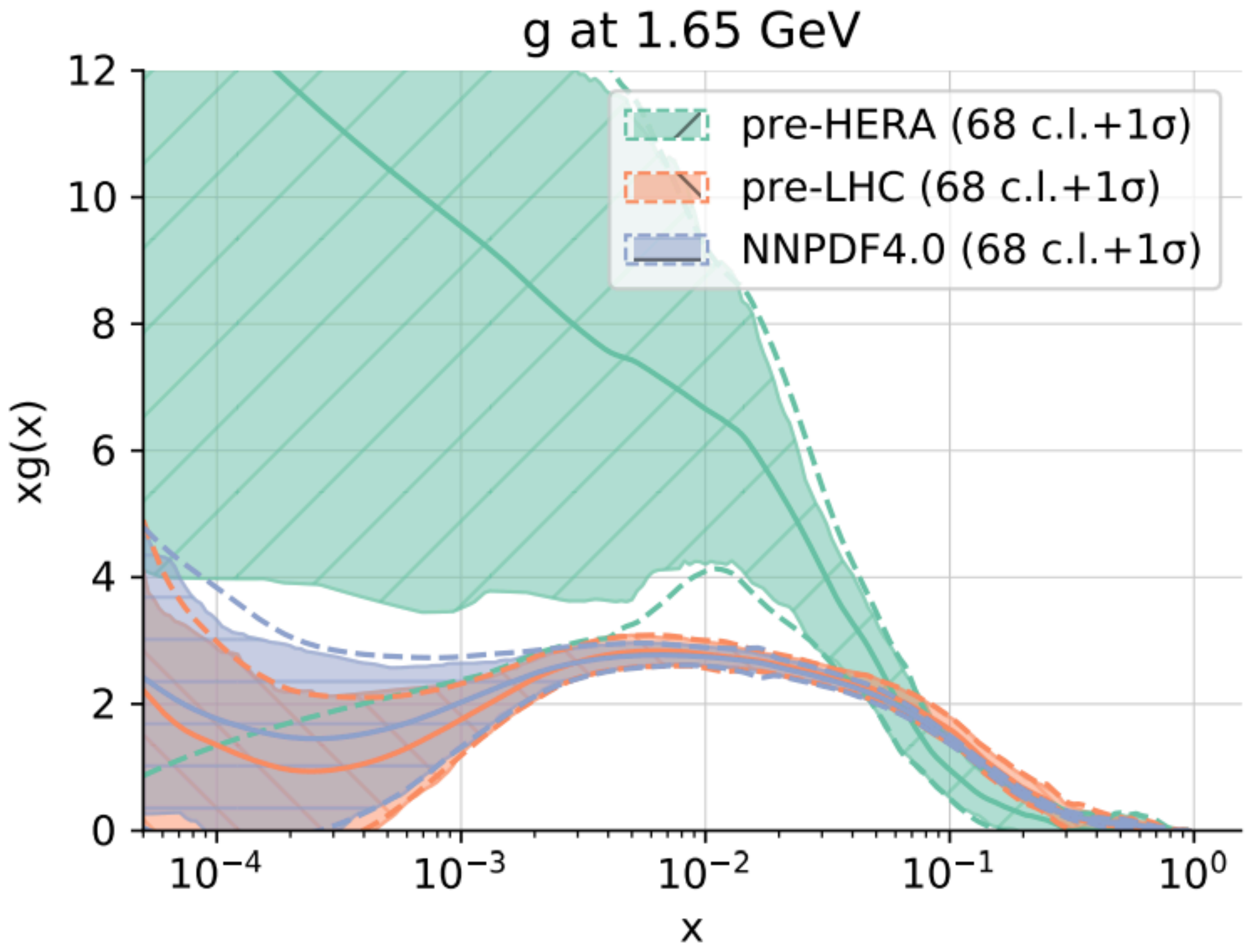
Future test

What about data you have not seen yet?

Traveling in time is not possible but I know **history!**

Divide the dataset **chronologically** and perform a fit for each set: **yesterday's extrapolation region is today's data region**



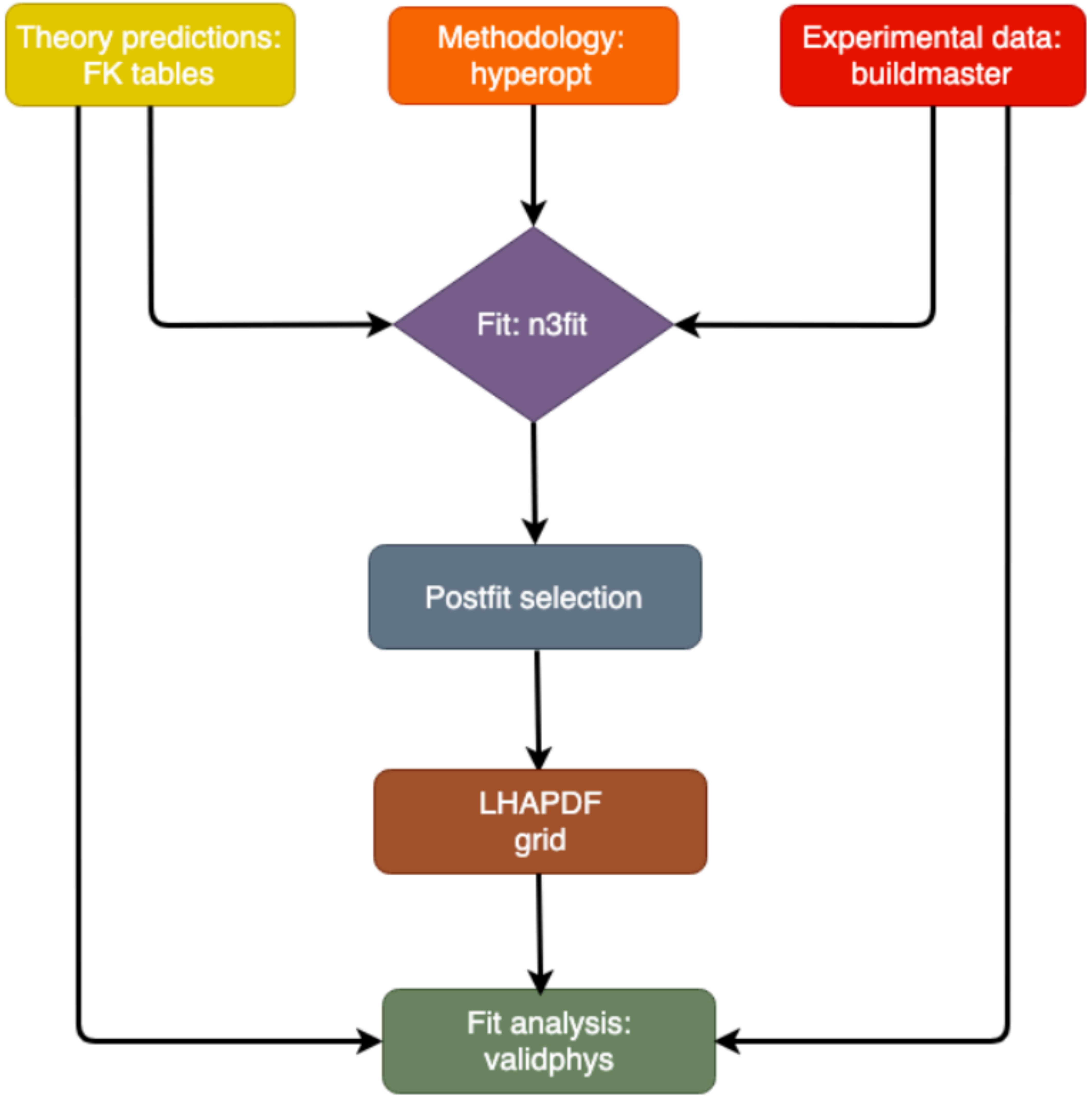


The NNPDF code is open-source

The full NNPDF code has been made **public** along with **user friendly documentation**

https://github.com/NNPDF/nnpdf

https://docs.nnpdf.science/



Outline

The Physics

The NNPDF framework

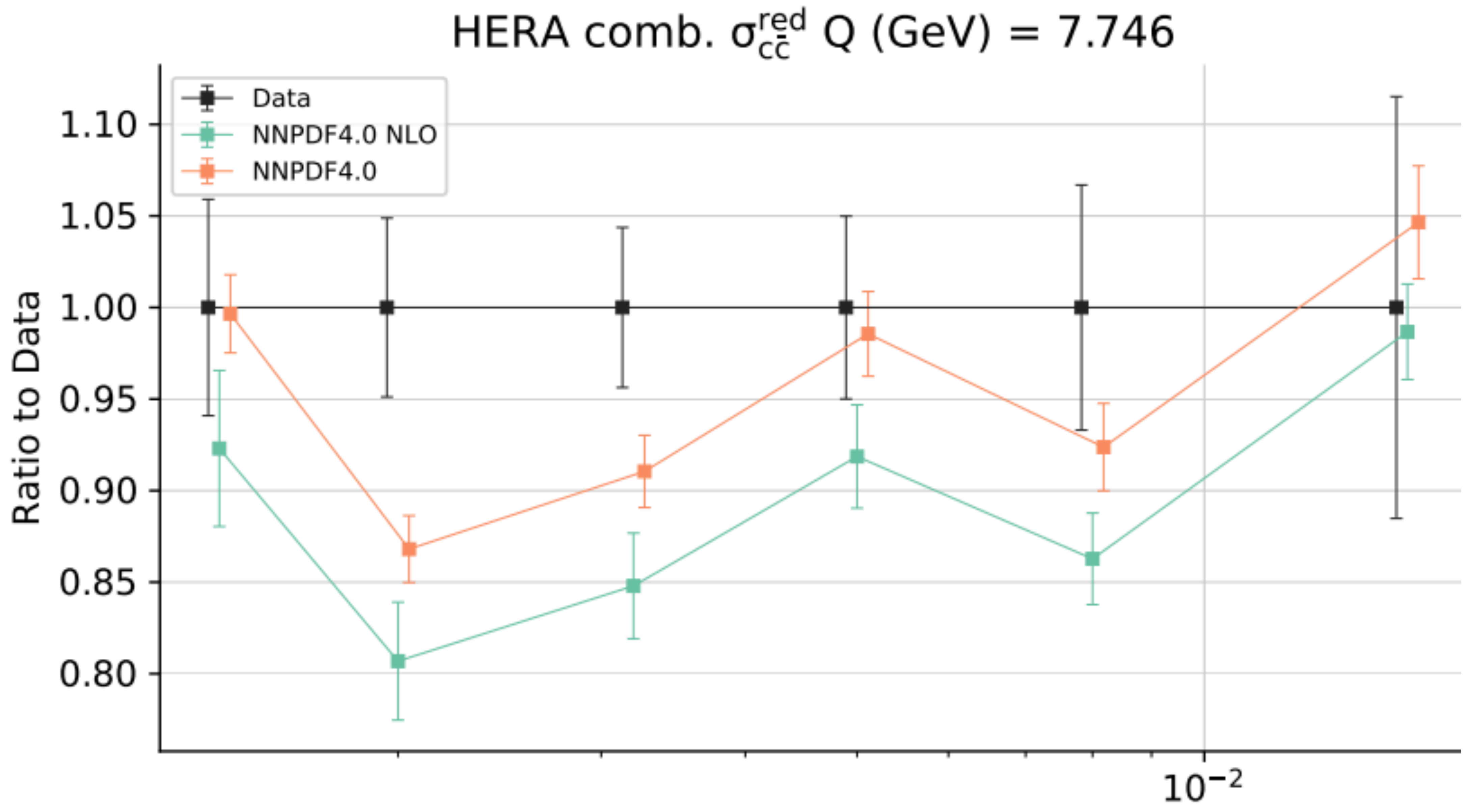
Results and outlook

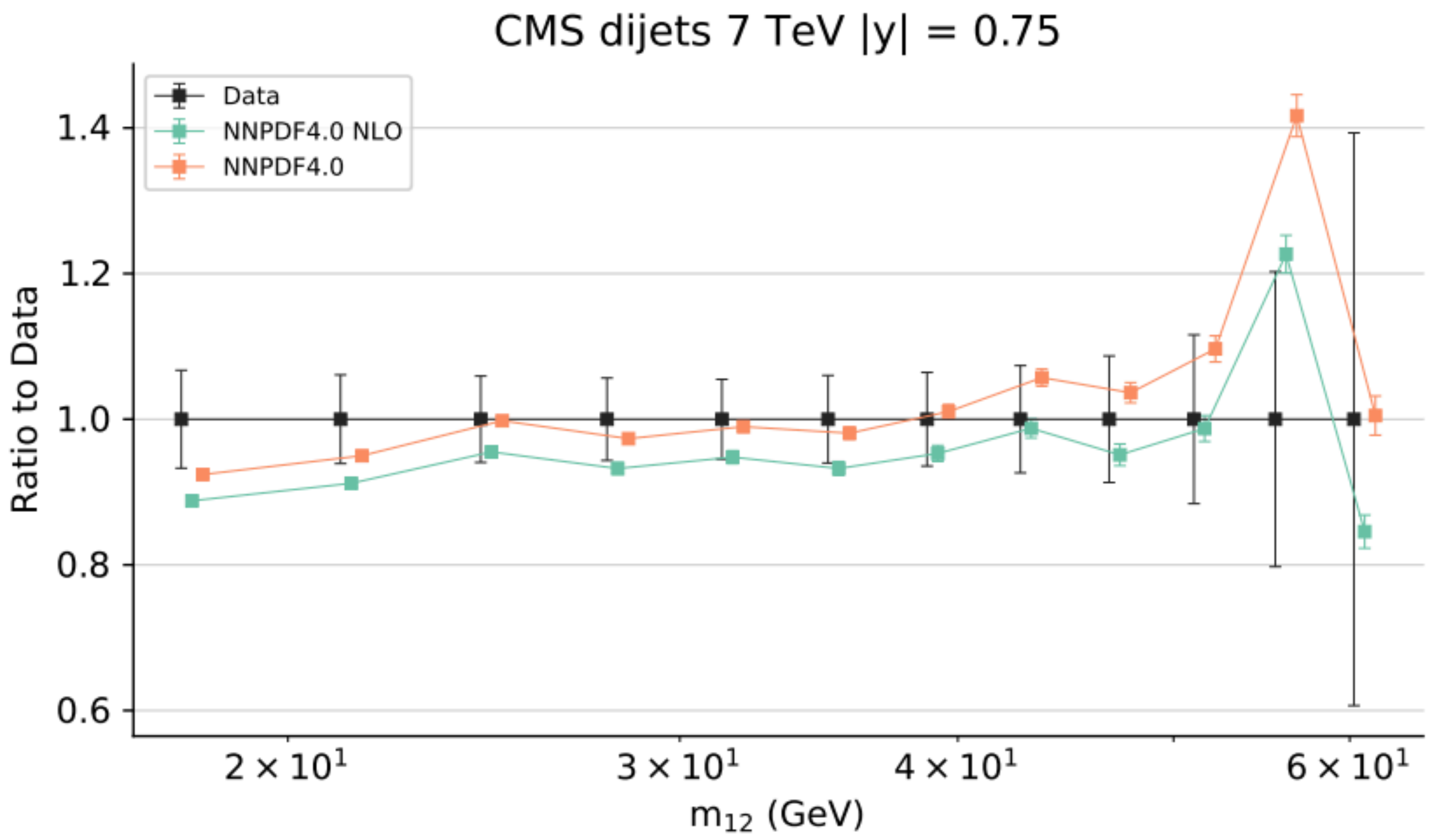
Fit quality: PDFs

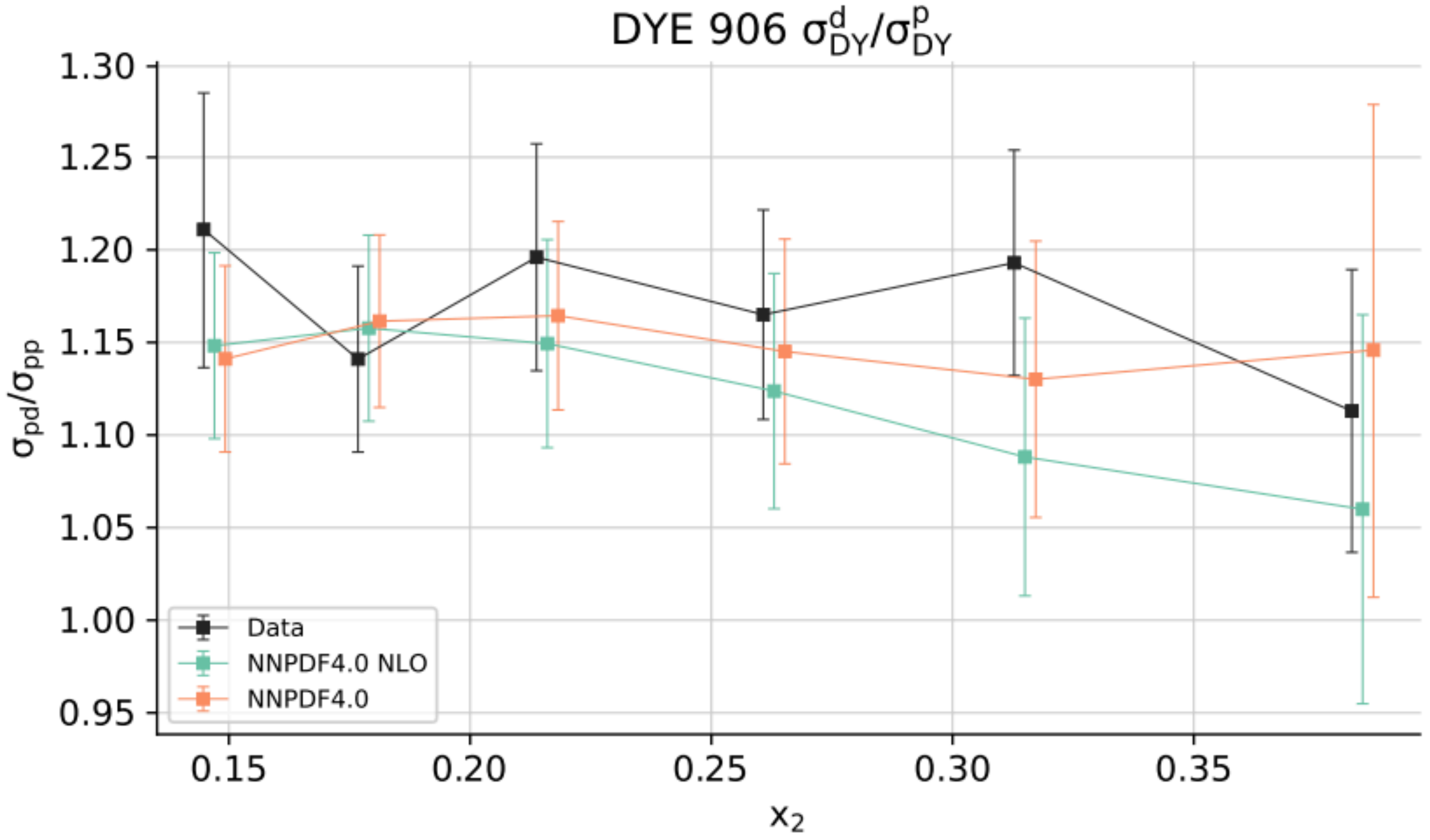
The fit quality clearly **improves** with the perturbative order (LO < NLO < NNLO)

Fit quality: predictions vs data

Also the description of the data clearly **improves** from NLO to NNLO





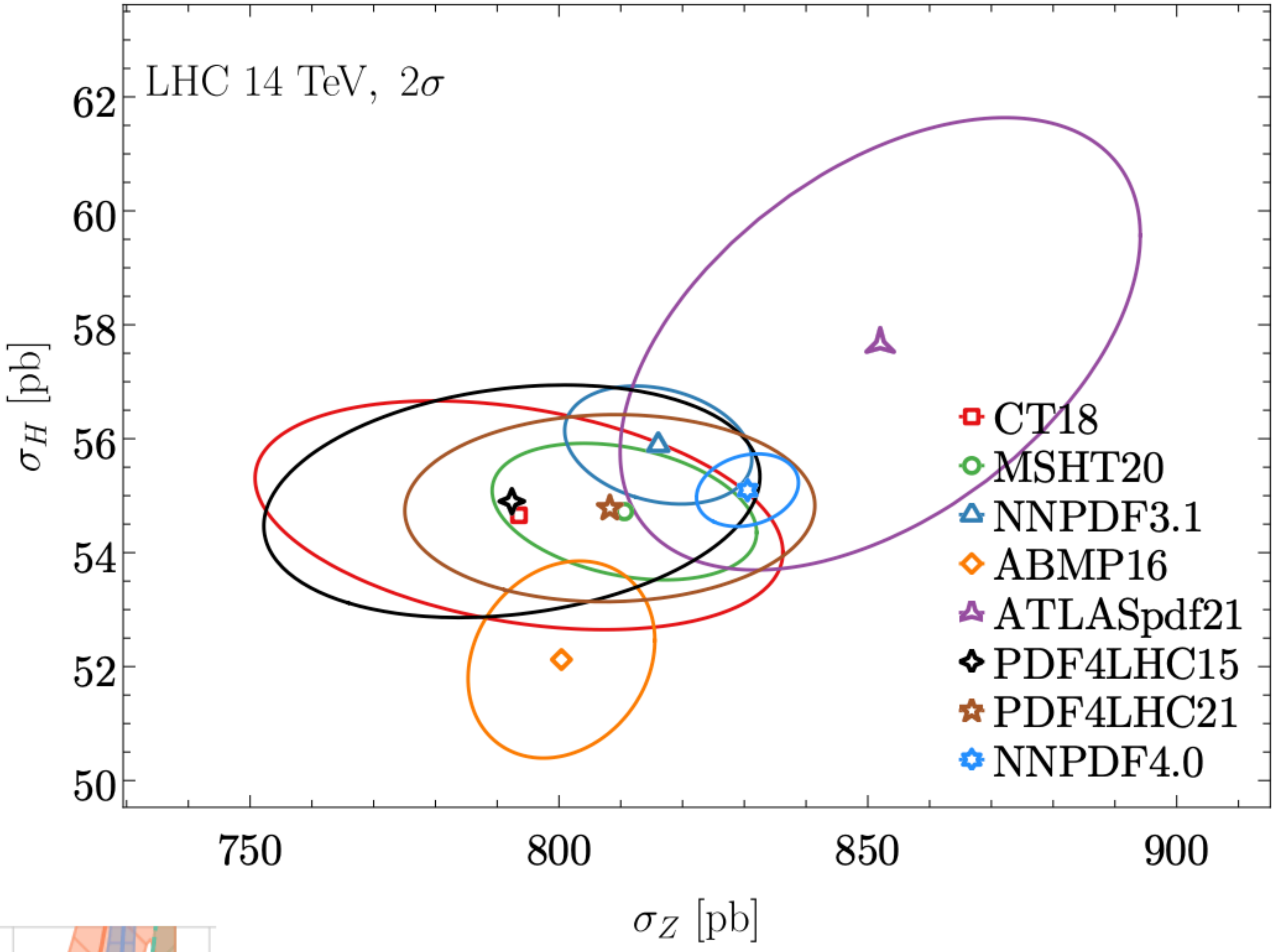


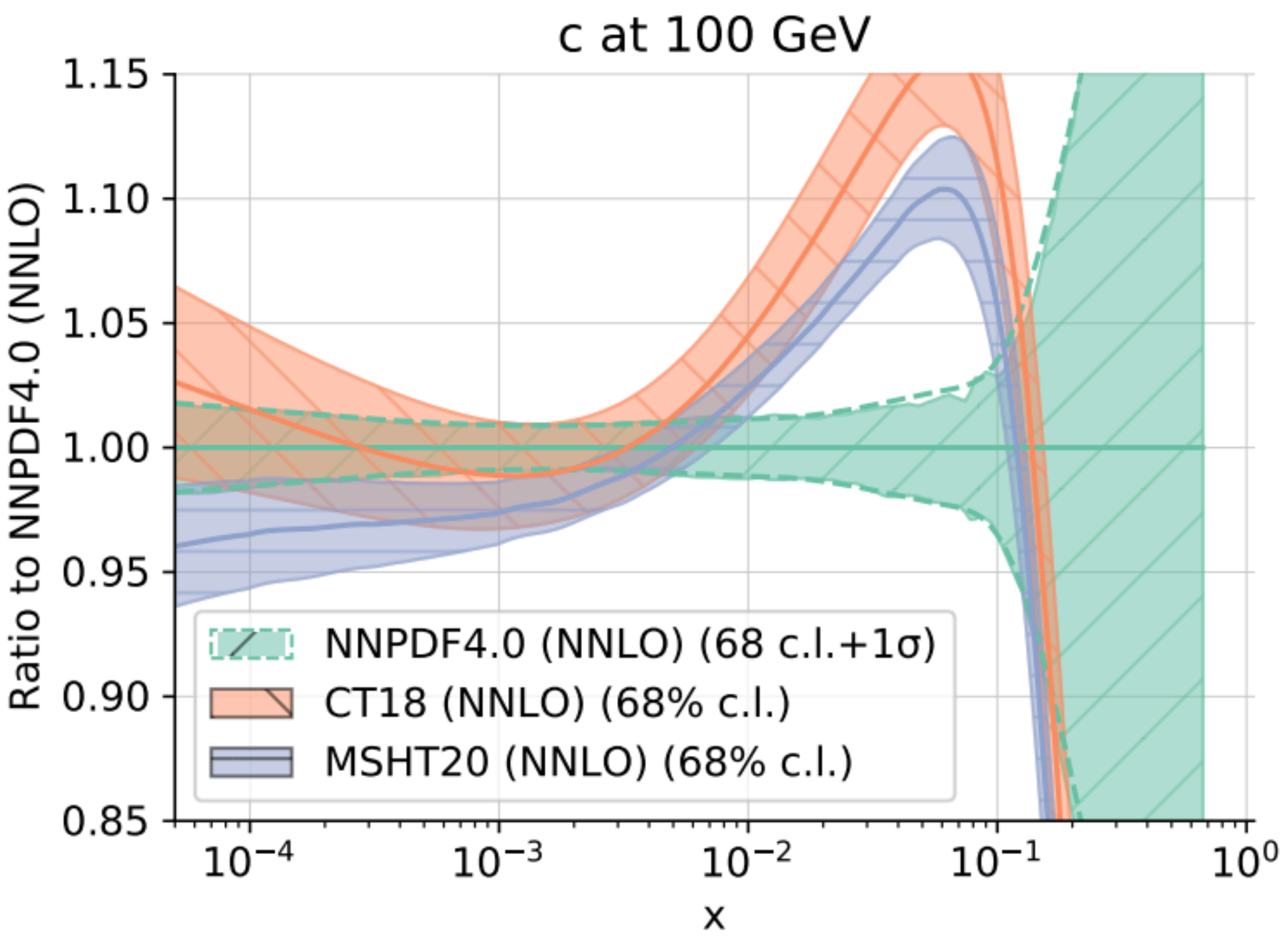
Comparison to other methodologies

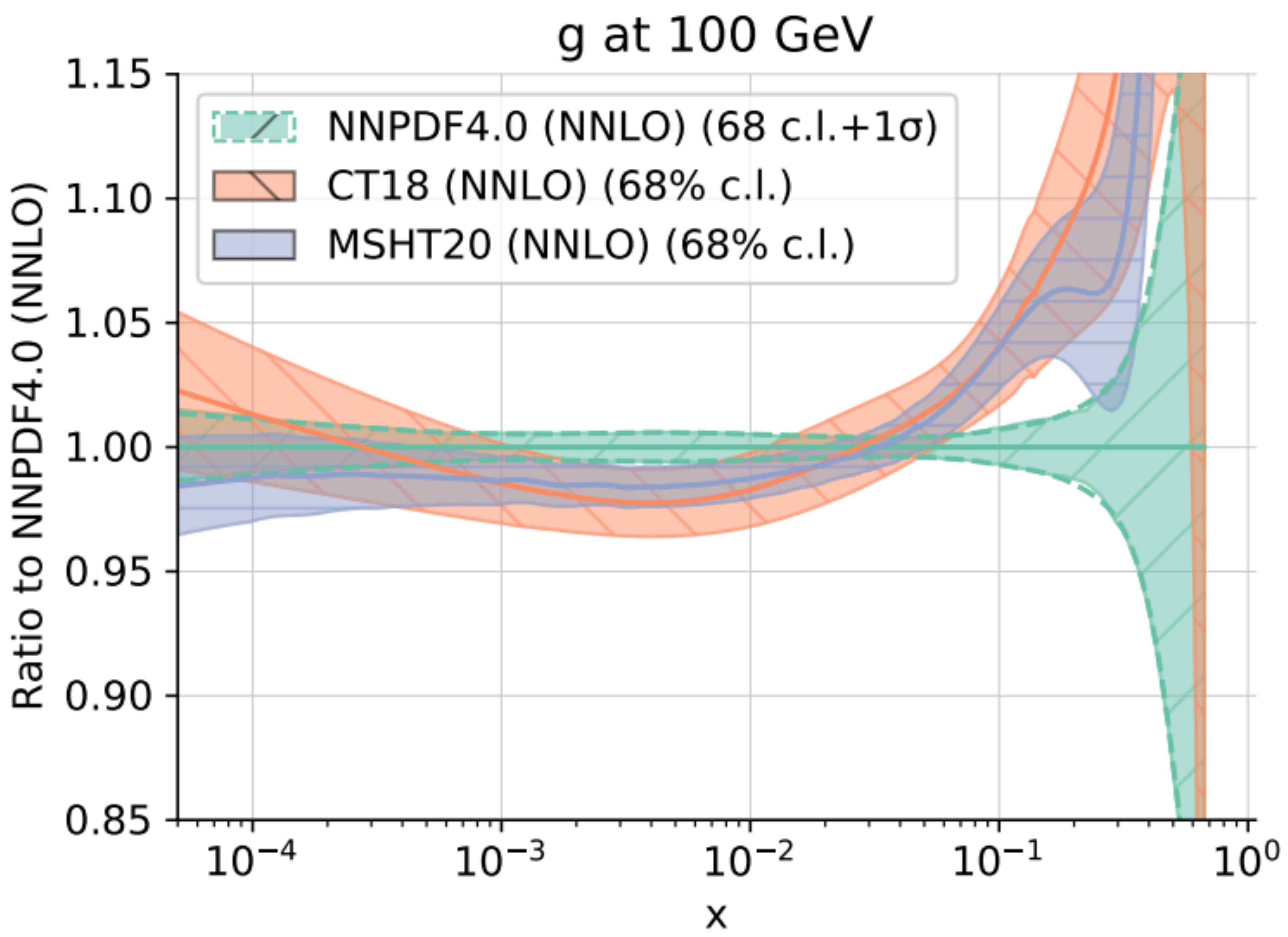
The agreement among different PDF fitting group is rather good

↓

However, NNPDF (3.1 and 4.0) has **smaller uncertainties** than the other groups → **effect of the NN**







Outlook: WIP and future projects

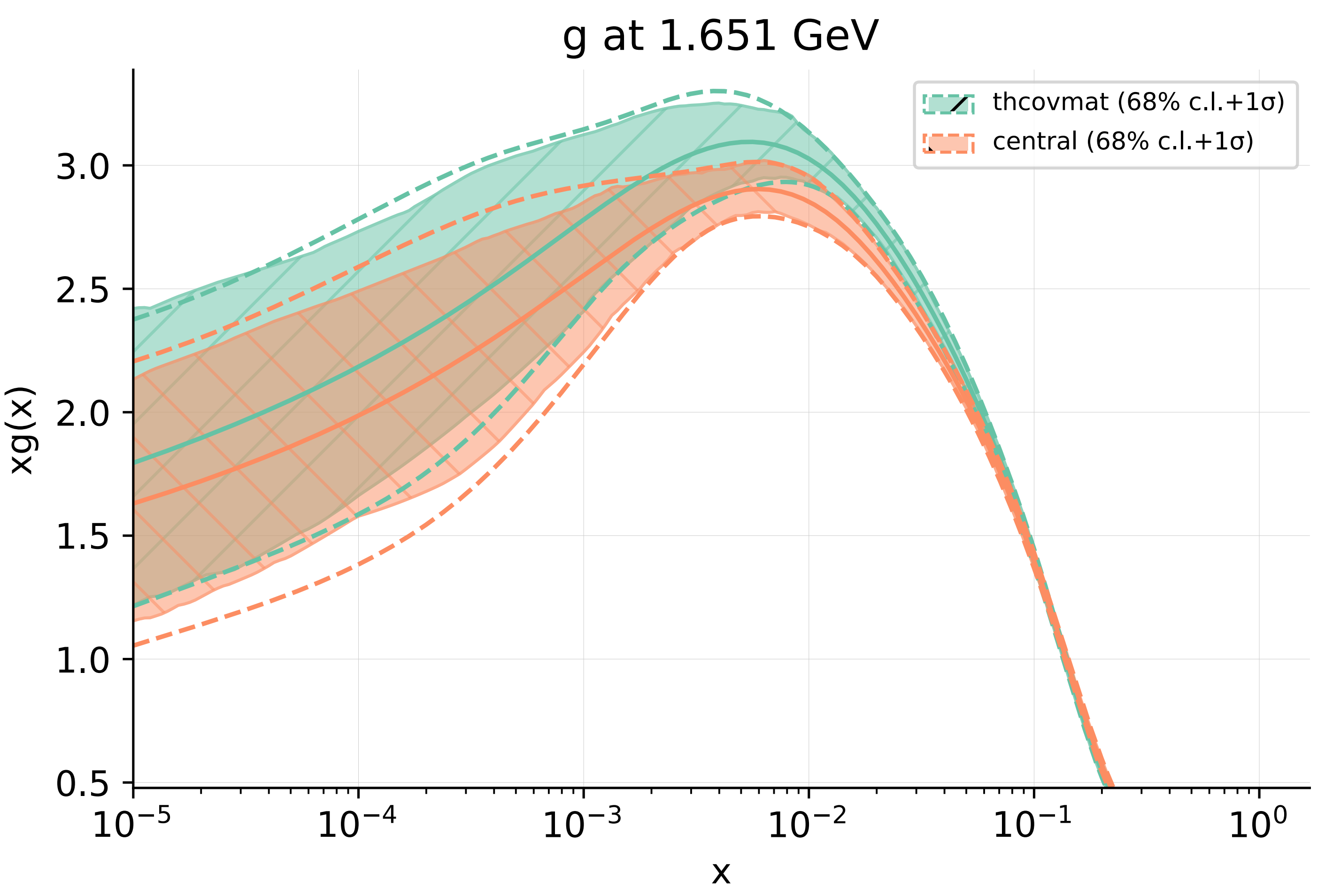
Physics projects

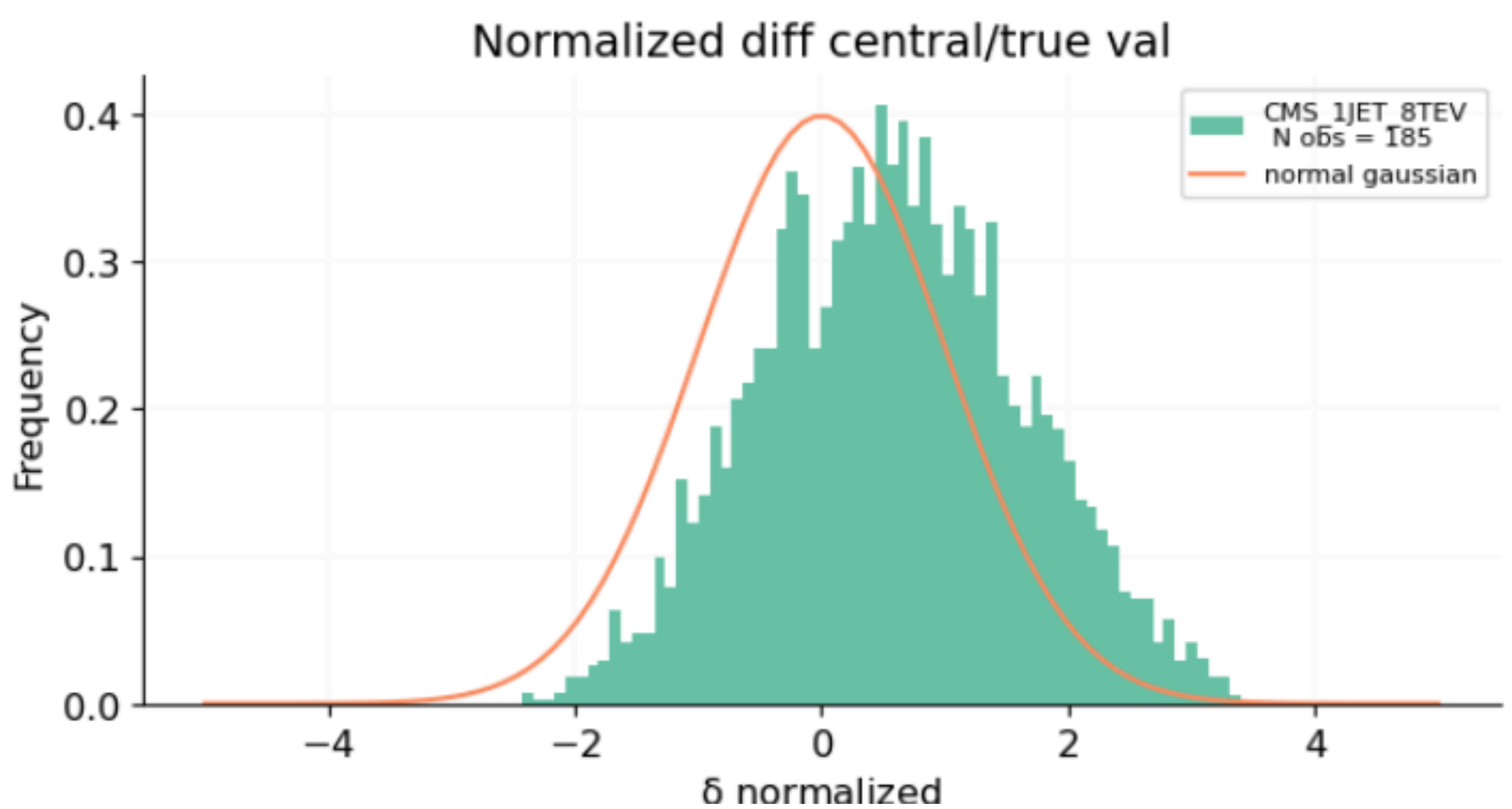
- Fit with theoretical uncertainties
- Fit with photon induced effects
- Fit at N3LO perturbative order

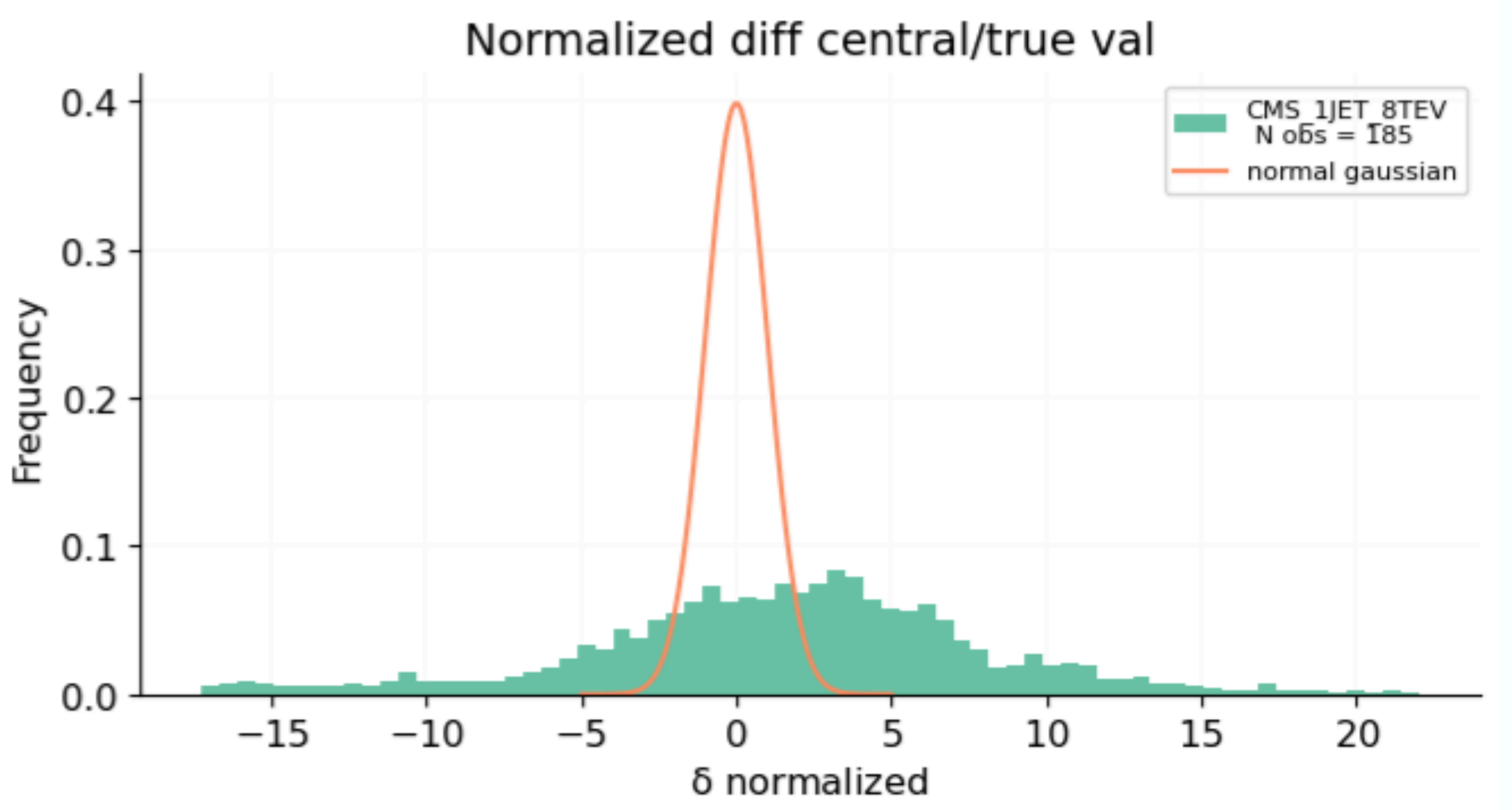
Methodology projects

- Bayesian fit
- New overfitting metrics for hyperopt
- Closure tests with inconsistent data

Preliminary results!







Conclusions

- Using neural networks techniques for PDF evaluation has led to several successes
- The comparison with other PDF fitting groups has shown that using NN techniques it is possible to obtain results with smaller uncertanties, while keeping them reliable
- For the future, it will be important to focus on explainability and improvements of the methodology

Neural networks techniques for parton distribution functions evaluation

Andrea Barontini on behalf of the NNPDF collaboration

Alpaca: modern algorithms in machine learning and data analysis: from medical physics to research with accelerators and in underground laboratories

20/11/2023