Machine Learning in PDF determination: NNPDF4.0

Juan M Cruz-Martinez

Eur.Phys.J.C 82 (2022); hep-ph/2109.02653

Transversity 2022, Pavia

European Research Council
Established by the European Commission

This project has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No 740006.

Outline

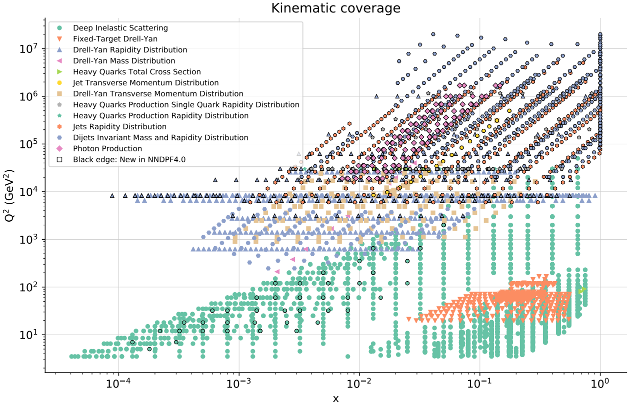
1. NNPDF4.0
   - The latest NNPDF set and methodology
   - Machine Learning for PDF determination
   - The NNPDF framework
2. NNPDF4.0 and beyond
   - Hyperoptimization: fitting the methodology
   - Handcrafting operations
   - Changing the backend
3. Conclusions

New PDF: new Data

NNPDF4.0 includes a plethora of new data

New processes:

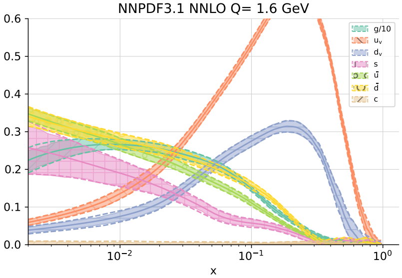
- direct photon
- single top
- dijets
- W+jet
- DIS jet



More than 4000 datapoints!

New PDF: new methodology

- ✓ **Stochastic Gradient Descent** for NN training using TensorFlow
- ✓ Automated optimization of **model hyperparameters**
- ✓ Methodology is validated using **closure tests** (data region), **future tests** (extrapolation region), and **parametrization basis independence**
- ✓ New and improved physical constraints: (**PDF positivity, integrability of nonsinglet distributions**)

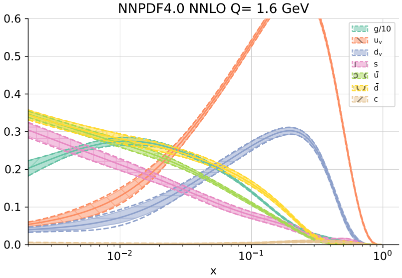


✓ **A completely open-source framework!**

In this talk the focus is on the NNPDF4.0 methodology

New PDF: new methodology

- ✓ **Stochastic Gradient Descent** for NN training using TensorFlow
- ✓ Automated optimization of **model hyperparameters**
- ✓ Methodology is validated using **closure tests** (data region), **future tests** (extrapolation region), and **parametrization basis independence**
- ✓ New and improved physical constraints: (**PDF positivity, integrability of nonsinglet distributions**)

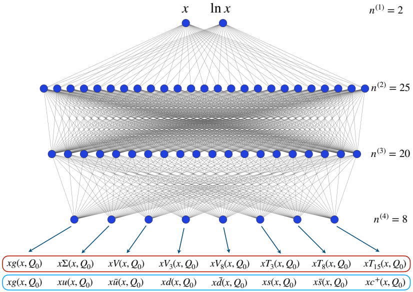


✓ **A completely open-source framework!**

In this talk the focus is on the NNPDF4.0 methodology

New PDF: new methodology

- ✓ **Stochastic Gradient Descent** for NN training using TensorFlow
- ✓ Automated optimization of **model hyperparameters**
- ✓ Methodology is validated using **closure tests** (data region), **future tests** (extrapolation region), and **parametrization basis independence**
- ✓ New and improved physical constraints: (**PDF positivity, integrability of nonsinglet distributions**)



$$f_i\,(x, Q_0) = x^{-\alpha_i}(1-x)^{\beta_i}\mathrm{NN}_i(x)$$

✓ **A completely open-source framework!**

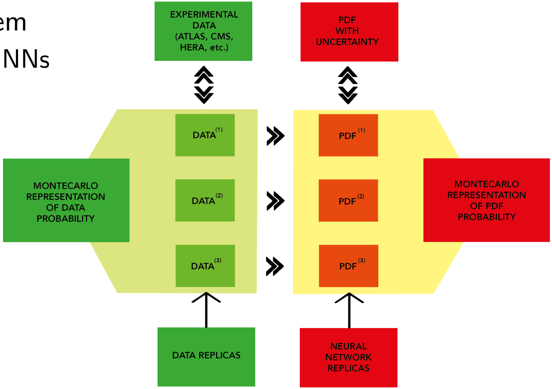
In this talk the focus is on the NNPDF4.0 methodology

PDFs as an ML problem: the NNPDF approach

Why use machine learning for PDF determination?

- ✓ Unknown functional form which needs to be inferred from data
- ✓ Well defined input and output
- ⇒ Supervised learning problem
  - PDFs parametrized by NNs

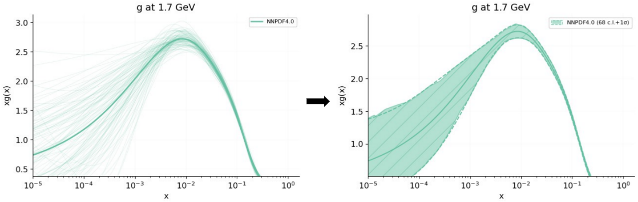
The NNPDF framework transforms distributions of experimental data into PDFs.



PDFs as an ML problem: the NNPDF approach

Why use machine learning for PDF determination?

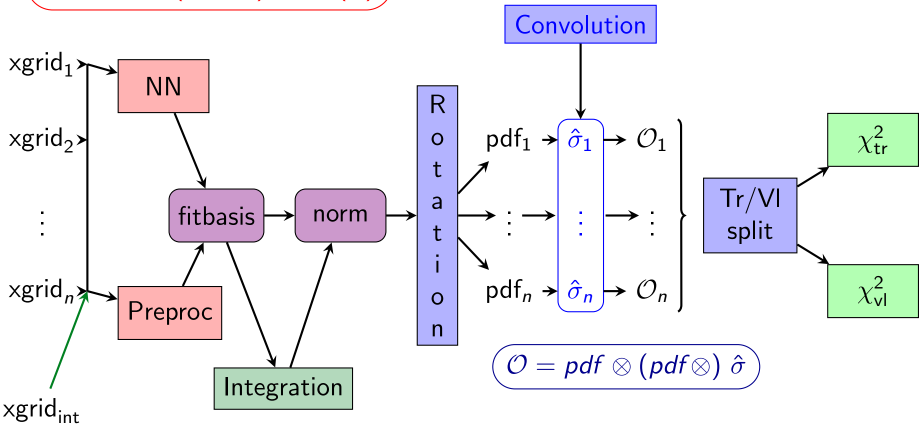
- ✓ Unknown functional form which needs to be inferred from data
- ✓ Well defined input and output
- ⇒ Supervised learning problem
  - PDFs parametrized by NNs



Replica sample of functions ⇔ Probability density of the PDF

How is that done in practice: The NNPDF model

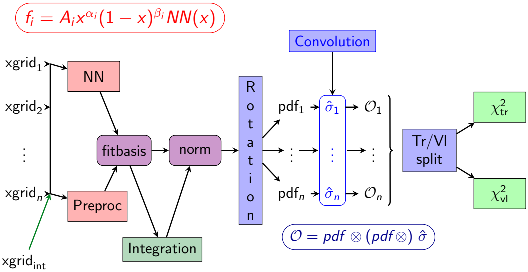
$$f_i = A_i x^{\alpha_i}(1-x)^{\beta_i} NN(x)$$



$$\mathcal{O} = pdf \otimes (pdf \otimes)\ \hat{\sigma}$$

NNPDF4.0 model

For more information see EPJ C79 (2019) 676



Main features:

- ✓ `Python` codebase: easier & faster development
- ✓ Object oriented for increased flexibility
- ✓ Freedom to use external libraries (default: `TensorFlow`)
- ✓ Modularity ⇒ can vary all aspects of the methodology

NNPDF framework: Eur.Phys.J.C 81 (2021) 10, 958; hep-ph/2109.02671

Where to obtain the code

The NNPDF framework is divided in the fitting code `n3fit` and the analysis toolbox `validphys` both of them available at: `github.com/NNPDF/nnpdf`

How to install

The NNPDF code can be easily installed using conda.

```
~$ conda install nnpdf -c https://packages.nnpdf.science/conda -c
  defaults -c conda-forge
```

Documentation

The documentation for the entirety of the code (fitting framework and analysis tools) is accessible at: docs.nnpdf.science

And... what can I do apart from reproducing NNPDF4.0?

NNPDF framework: Eur.Phys.J.C 81 (2021) 10, 958; hep-ph/2109.02671

Where to obtain the code

The NNPDF framework is divided in the fitting code `n3fit` and the analysis toolbox `validphys` both of them available at: `github.com/NNPDF/nnpdf`

How to install

The NNPDF code can be easily installed using conda.

```
~$ conda install nnpdf -c https://packages.nnpdf.science/conda -c
  defaults -c conda-forge
```

Documentation

The documentation for the entirety of the code (fitting framework and analysis tools) is accessible at: docs.nnpdf.science

Anything you want-*ish*

Performance benefit - time per replica

	NNPDF3.1	NNPDF4.0 (CPU)	NNPDF4.0 (GPU)
Time p/replica	15.2 h	38 min	6.6 min
Speed up	1	24	140

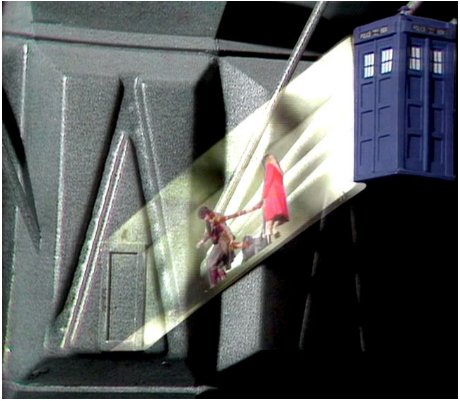
- Fewer CPU hours for a fit
- Use of gradient descent optimization $\Rightarrow$ more stable results

$\Rightarrow$ Scan over thousands of hyperparameter combinations and select the best one

$\Rightarrow$ Possible to automatically learn the methodology

The art of the hyperparameter selection

Just as technology has changed the way movies are made, one of studies that the new code enables, is the automatic and systematic **hyperparameter scan** which is rendered possible by the advances in technology and the new code's speed.



1978

2014

Beyond the PDF fit: fitting the methodology

The main objective of NNPDF is to minimize choices that can bias the PDF:

- ✗ Functional form $\longrightarrow$ Neural Networks
- ✗ However: NN are defined by set of parameters!

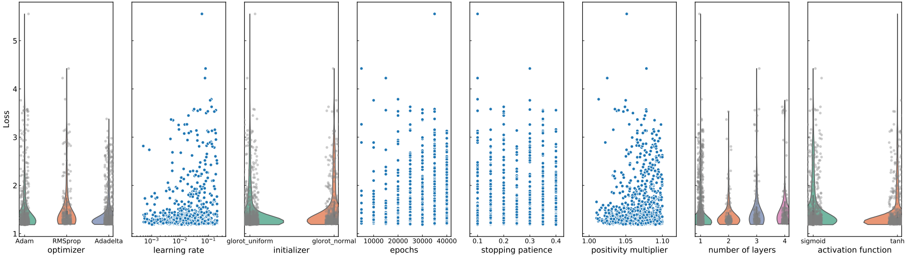
Humans are good at recognising patterns but selecting the best set of parameters is a slow process and systematic success is not guaranteed

To overcome this selection problem we implement a hyperparameter scan: let the computer decide automatically

- ✓ Scan over thousands of hyperparameter combinations
- ✓ Define a reward function to grade the model
- ✓ Check the generalization power of the model

Hyperparameter scan

Each blue dot corresponds to a fit of a different set of hyperparameters:



Thousands of fits for the hyperoptimization algorithm to choose:

- ✓ Optimizer
- ✓ Initializer
- ✓ Stopping Patience
- ✓ Number of Layers
- ✓ Learning Rate
- ✓ Epochs
- ✓ Positivity Multiplier
- ✓ Activation Function

Hyperoptimization: reward and generalization

If we use as hyperoptimization target the χ^2 of the fitted data, we risk finding the hyperparameter set that better overfits.

We avoid this problem by adopting ***k*-folding**:

- Divide the data into k sets.
- Leave one set out and fit the $k-1$ sets left.
- Optimize the average χ^2 of the k non-fitted sets.

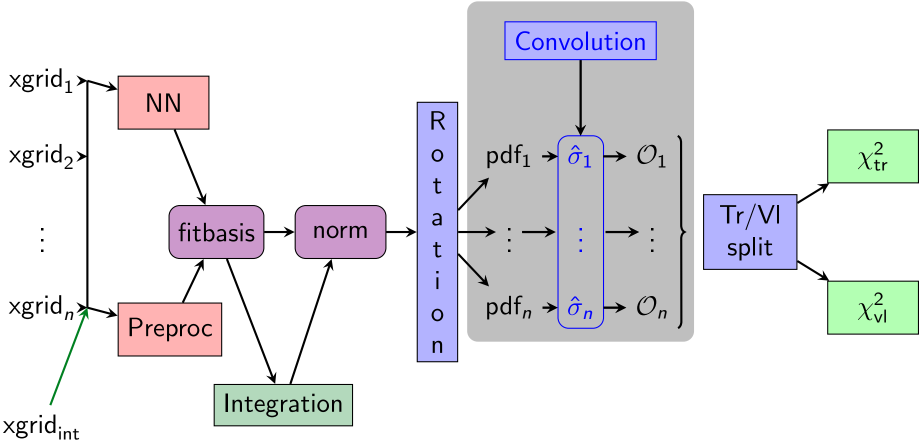
Example of function to hyperoptimize:

$$\text{Loss}(optimizer\_name,\ depth\_of\_network) = \frac{1}{k}\sum_{k}^{i} \frac{\chi_i^2}{N_i}$$

Where we are computing the χ^2 for the data that did not enter the fit. This ensures that the methodology can accommodate well even data that has never been seen by the fit.

Customizing the operations

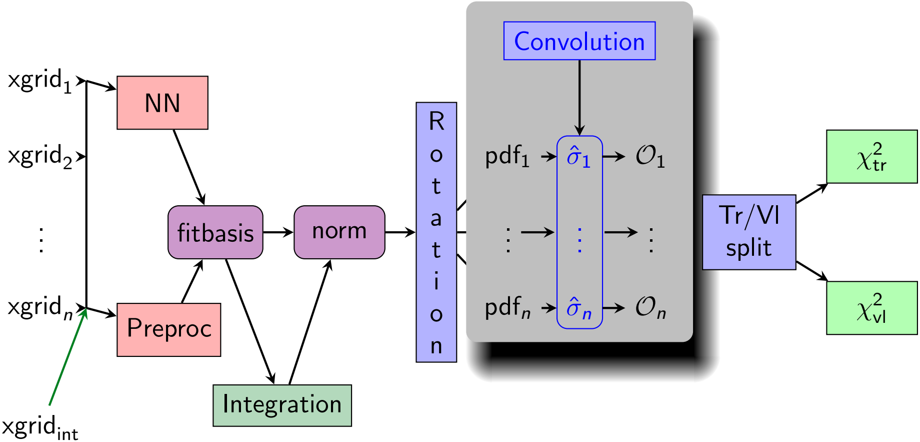
Tensorflow is very clever, but we have more information:



Customizing the operations

Tensorflow is very clever, but we have more information:

$\longrightarrow$ It is possible to hand-craft our own operators



Customizing the operations

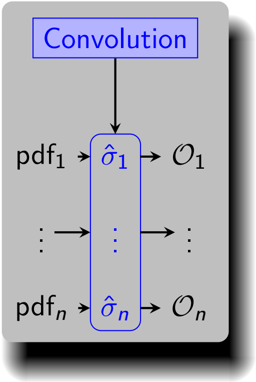
Tensorflow is very clever, but we have more information:

$\longrightarrow$ It is possible to hand-craft our own operators

	TensorFlow	Our own
Memory Total	18.4 Gb	12.5 Gb
Memory Fit	16.3 Gb	10.4 Gb

Timings are similar between the hand-crafted and the default TF convolution

As the memory is reduced we can "fit" more and more replicas in one single run: time reduction is a function of the memory.

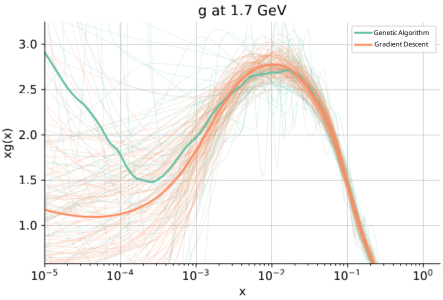


Going back to Genetic Algorithms

PoS AISIS2019 (2020) 008; physics.comp-ph/2002.06587

TensorFlow contains only gradient-descent based algorithms, if we want to again use Genetic Algorithms, we would need to modify the backend!

- ✓ The flexibility of the NNPDF framework allows to change the optimizer
- ✓ Doesn't even need to be `TensorFlow` or `python` based!

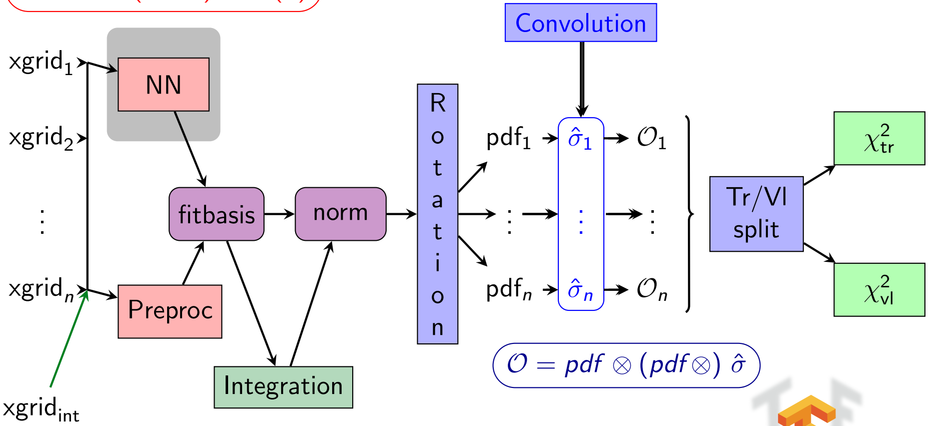


Everything else remains the same, we only need to change the exact piece we want to modify!

Using a Quantum Computer to simulate PDFs: QPDF

Phys.Rev.D 103 (2021) 3, 034027; hep-ph/2011.13934

$$f_i = A_i x^{\alpha_i}(1-x)^{\beta_i} NN(x)$$



$$\mathcal{O} = pdf \otimes (pdf \otimes)\ \hat{\sigma}$$

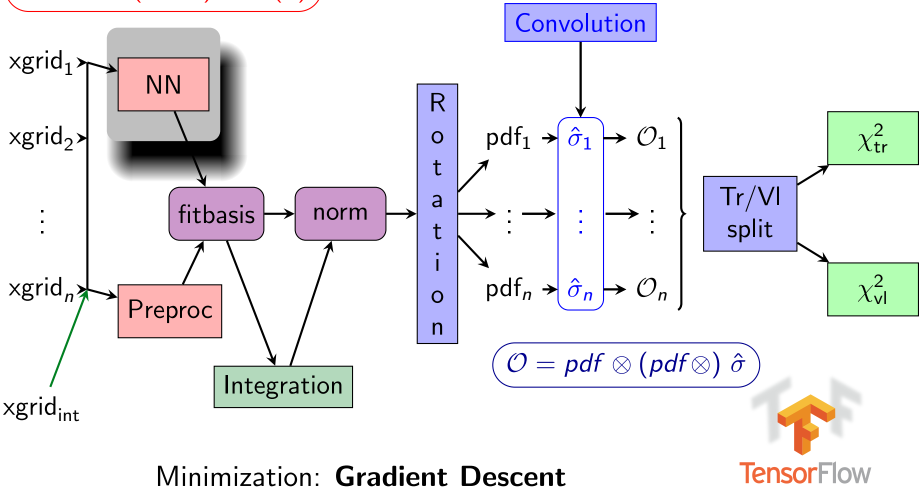
Minimization: **Gradient Descent**

TensorFlow

Using a Quantum Computer to simulate PDFs: QPDF

Phys.Rev.D 103 (2021) 3, 034027; hep-ph/2011.13934

$$f_i = A_i x^{\alpha_i}(1-x)^{\beta_i} NN(x)$$

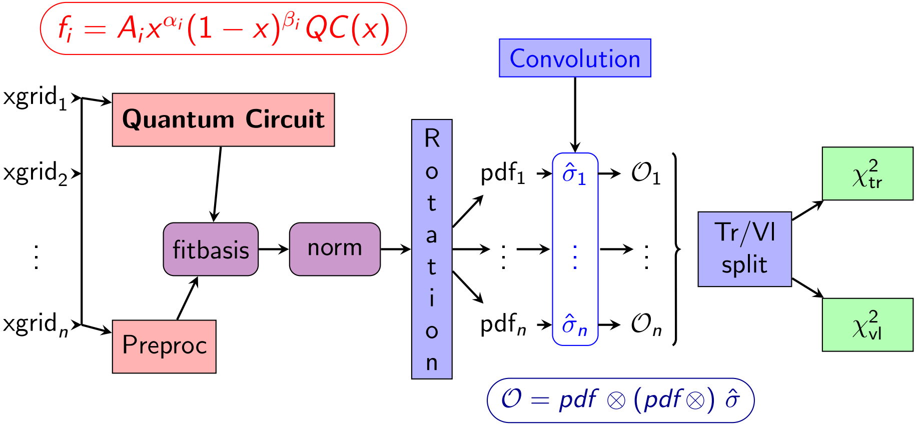


$$\mathcal{O} = pdf \otimes (pdf \otimes)\ \hat{\sigma}$$

Minimization: **Gradient Descent**

Using a Quantum Computer to simulate PDFs: QPDF

Phys.Rev.D 103 (2021) 3, 034027; hep-ph/2011.13934



Minimization: **BFGS**

Summary

✓ **NNPDF 4.0:** The latest set of NNPDF PDFs is both more accurate *and* precise (many checks to test both!)

✓ NNPDF machinery for PDF fitting is faster, flexible and more powerful.

✓ The framework allows for full customization *by design*.

Where to check the documentation?

NNPDF is documented at `docs.nnpdf.science`

Where to obtain the code?

NNPDF is open source and available at `github.com/NNPDF/nnpdf`

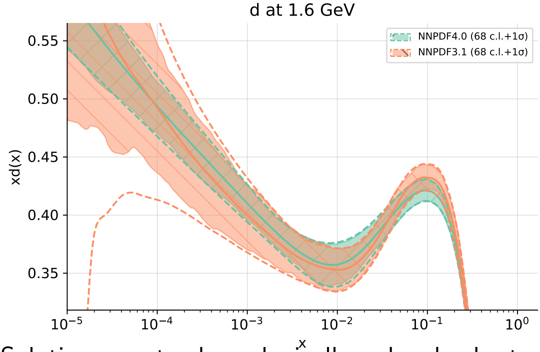
If you have any question about the usage of the framework just open an issue in the repository or drop me an email, we are always happy to help!

Thanks!

How can future-proof the methodology

Do we trust our errorbands?

The smaller error bands in the NNPDF4.0 fits are driven both by the increased amount of data and the improved methodology.



Ideally: design an experiment for the regions not covered by fitted-data!

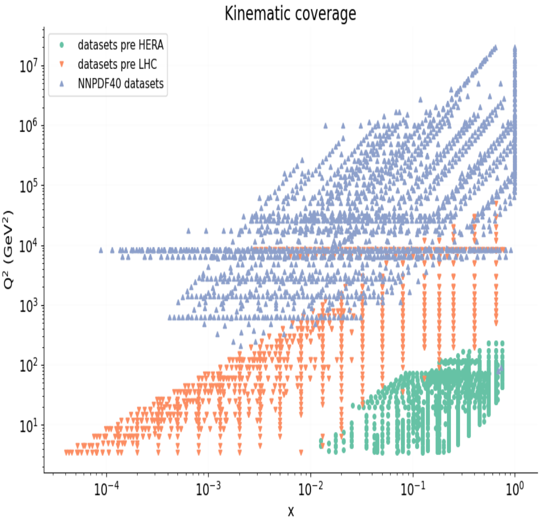
Problem: we want the results before 2050...

Figure: Other valid and certified future-testing methods

Solution: create chronologically ordered subsets of data and check the methodology in each of these situations, we call this “future tests“.

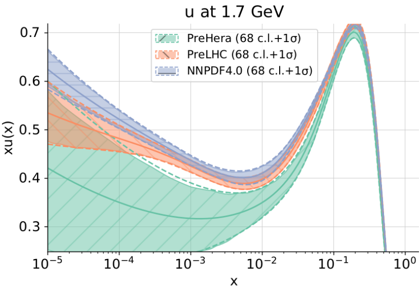
Future tests

for more information see arxiv:2103.08606



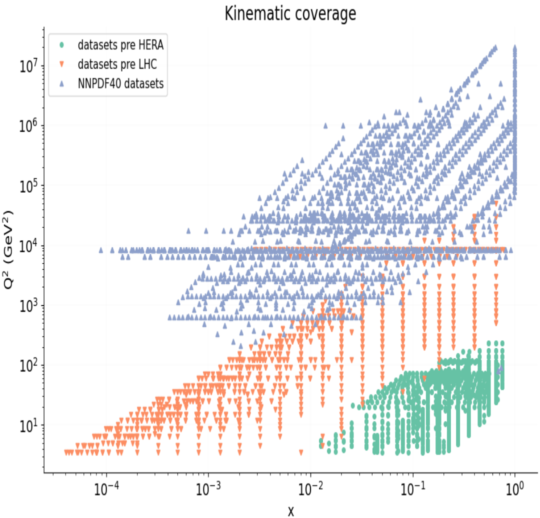
χ^2/N (only exp. covmat)

(dataset)	NNPDF4.0	pre-LHC	pre-Hera
pre-HERA	1.09	1.01	0.90
pre-LHC	1.21	1.20	**23.1**
NNPDF4.0	1.29	**3.30**	**23.1**



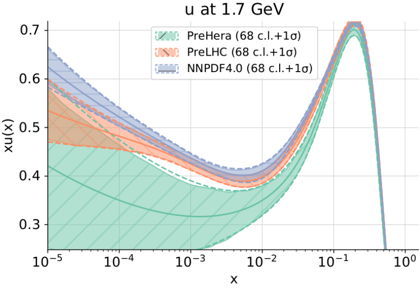
Future tests

for more information see arxiv:2103.08606

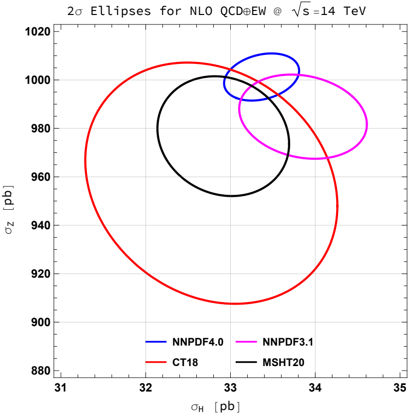


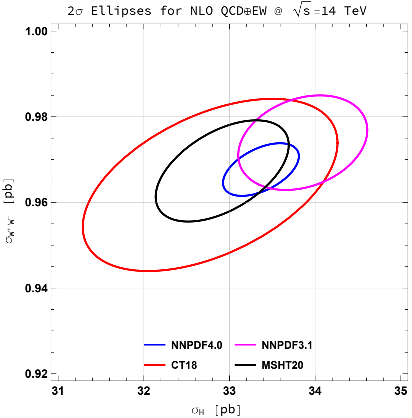
χ^2/N (exp. and PDF covmat)

(dataset)	NNPDF4.0	pre-LHC	pre-Hera
pre-HERA			0.86
pre-LHC		1.17	**1.22**
NNPDF4.0	1.12	**1.30**	**1.38**



PDF uncertainties of different PDF sets





NNLO theoretical predictions for 95% C.L. PDF uncertainties for several cross section values. Plot by T. Rabemananjara.