

# Studying the parton content of the proton with deep learning models

Juan M Cruz-Martinez



Symposium Artificial Intelligence for Science, Industry and Society  
Mexico City - 2019



**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 Introduction
  - References
  - Parton Distribution Functions: PDF
  - NNPDF
- 2 A new methodology, codename `n3fit`
  - Speed & flexibility → more physics
  - Hyperoptimization: fitting the methodology
  - New methodology, new fits
- 3 Conclusions

# NNPDF & N3PDF



[n3pdf.mi.infn.it](http://n3pdf.mi.infn.it)



[nnpdf.mi.infn.it](http://nnpdf.mi.infn.it)

The first main result of the N3PDF group is the new fitting code: `n3fit` which will be used for the forthcoming NNPDF4.0 PDF set.

→ A publication detailing the methodology here presented is found at  
[Eur.Phys.J. C79 \(2019\) no.8, 676](#) (S. Carrazza, **JCM**).

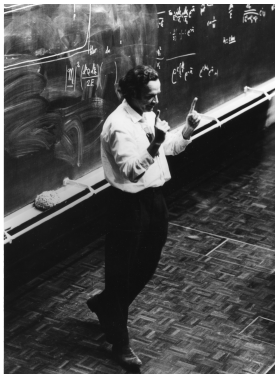
→ A conference proceedings detailing progress in hardware acceleration of PDF fitting can be found at

[hep-ph/1909.10547](#) (S. Carrazza, **JCM**, J. Urtasun-Elizari, E.Villa)

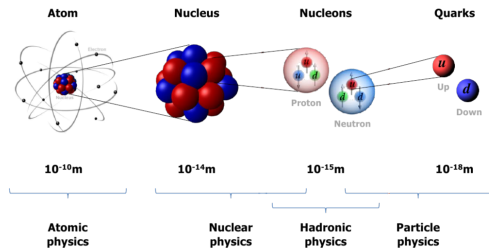
## History: the parton model

In the early days of particle physics the nucleons (protons and neutrons) were thought to be fundamental particles.

By the 70s however, hadrons moving at high speeds were modeled as collections of particles with some wide-spread momentum distribution.



Pic: Richard Feynman



Quarks and gluons within the hadrons are collectively known as partons.



## History: the parton model

In the early days of particle physics the nucleons (protons and neutrons) were thought to be fundamental particles.

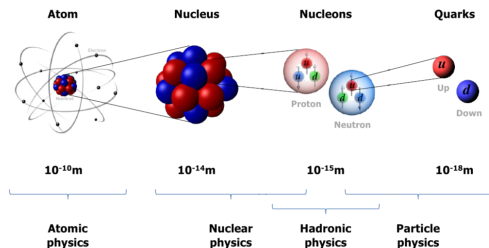
By the 70s however, hadrons moving at high speeds were modeled as collections of particles with some wide-spread momentum distribution.

$$\text{PDF} = f(x, Q^2)$$

$f$ :  
unknown function

$Q^2$ :  
energy of the proton

$x$ :  
fraction of  $Q^2$  carried by  
the parton



Partons within the proton are characterized by a probability density function of the hadron momentum: the PDF.

# The Parton Model

- PDFs are essential for the computation of physical observables involving hadrons, such as the cross section, as they characterize the input particles.

$$\mathcal{O} = \int_0^1 dx_1 dx_2 f_1(x_1) f_2(x_2) \hat{\sigma}(x_1, x_2) = \hat{\sigma} \otimes \text{PDF}$$

- Cannot currently be computed from first principles with current methods: low energy QCD interactions within the proton cannot be perturbatively described.
- PDFs are calculated experimentally by comparing theoretical predictions to real data.

A review on PDFs for those who are not familiar with the topic by Prof. Juan Rojo can be found at [hep-ph/1910.03408](https://arxiv.org/abs/hep-ph/1910.03408)

# Fitting the PDF

Historically, the PDF were fitted via relatively simple functional forms (polynomials). Available experimental data was also limited.

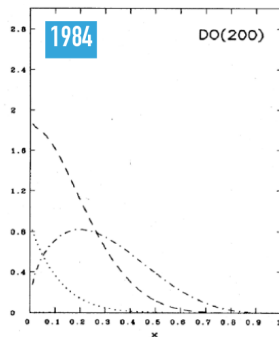
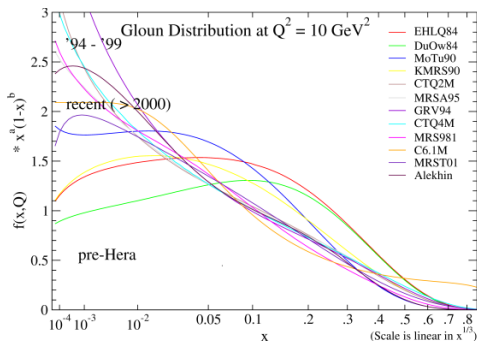


FIG. 27. "Soft-gluon" ( $A=200$  MeV) parton distributions of Duke and Owens (1984) at  $Q^2=5$  GeV<sup>2</sup>: valence quark distribution  $x[u_v(x)+d_v(x)]$  (dotted-dashed line),  $xG(x)$  (dashed line), and  $q_v(x)$  (dotted line).



New techniques have been developed and new experimental data has been obtained → which leads to better PDFs.

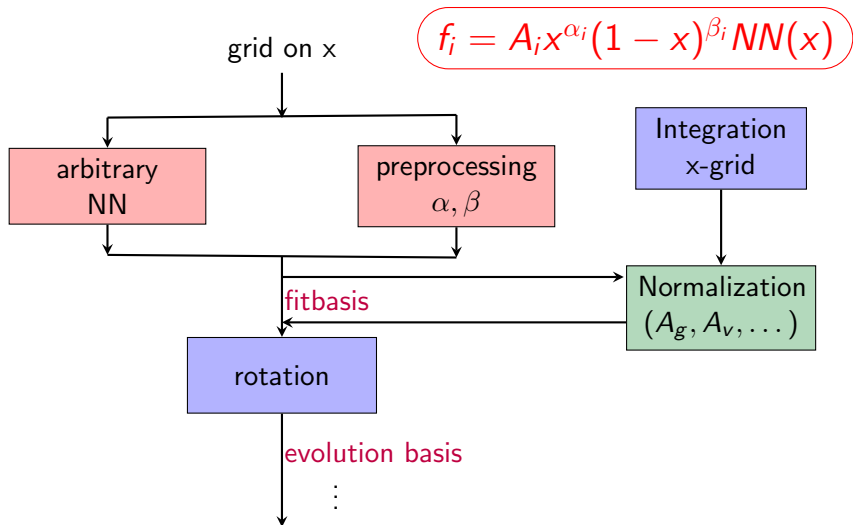
# PDF determination with Machine Learning

PDF determination is a problem very well suited for Machine Learning techniques:

- The functional form is not known.
  - Reduce theoretical bias
- It is at its heart an optimization problem
  - Figure of merit:  $\chi^2$  of the fit.
- Very well defined input and output
  - Input:  $x$  of the parton
  - Output: experimental data

The NNPDF collaboration introduced, alongside some other novel methodological developments, Neural Networks as a way to parametrize the functional form of the PDFs.

## Schematic view of the NNPDF PDF model



# The Loss Function

The fitting strategy is based on the minimization of the  $\chi^2$  of the fit,

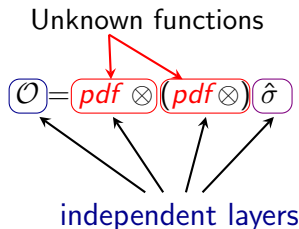
$$\chi^2 = \frac{1}{N} \sum (\mathcal{O}^i - \mathcal{D}^i) \sigma_{ij}^{-1} (\mathcal{O}^j - \mathcal{D}^j)$$

$N$ : number of data points

$\mathcal{O}^i$ : theoretical prediction

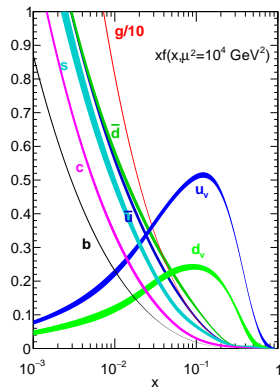
$\mathcal{D}^i$ : experimental data point

$\sigma_{ij}$ : covariance matrix



Note: The partonic cross section  $\hat{\sigma}$  correspond to APFELgrid tables as described in [hep-ph/1605.02070](https://arxiv.org/abs/hep-ph/1605.02070)

# The NNPDF methodology



Some key points of the technology used in the 3.1 release which are addressed in this talk:

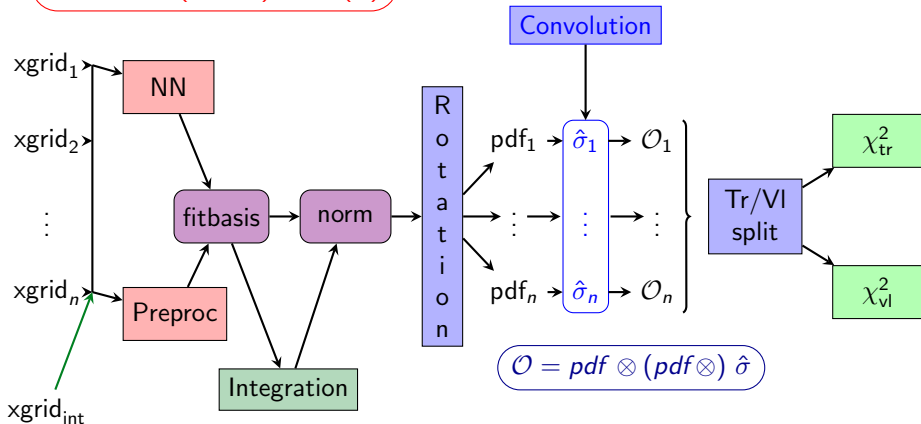
- NN are optimized with Genetic Algorithms
- All code is in-house (there was no Tensorflow 15 years ago!)
- Fit parameters are chosen manually

For a detailed review on the current NNPDF methodology please see the release paper of NNPDF 3.1: [Eur. Phys. J. C77 2017, 10](#)

We present now `n3fit`, developed within the NNPDF collaboration by the N3PDF team with the view of further improve PDF determination.

# The full n3fit model

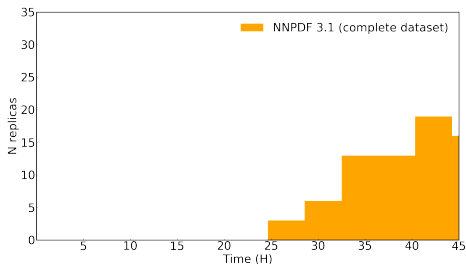
$$f_i = A_i x^{\alpha_i} (1-x)^{\beta_i} NN(x)$$





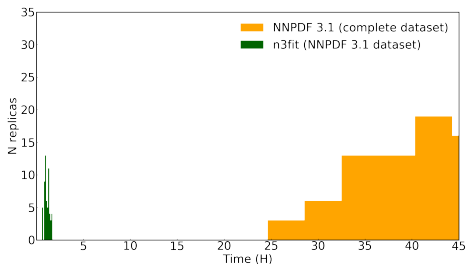
# A new generation of PDF fitting

- ✓ Rationalization of development wrt to older codes
  - Easier and faster development
  - Object orientation: full freedom and flexibility
- ✓ Gains on speed and efficiency:
  - Less CPU hours for a fit
  - Usage of new technologies
    - ✓ New hardware
    - ✓ New libraries
  - Usage of Gradient Descent methods
- ✓ Consequences
  - **Speed-up of research:** faster to develop, test, run
  - **More studies available**
  - Example: **fitting the methodology** (hyperparameter scan)



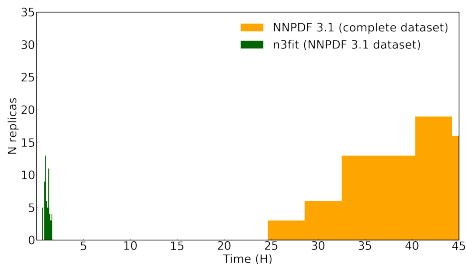
# A new generation of PDF fitting

- ✓ Rationalization of development wrt to older codes
  - Easier and faster development
  - Object orientation: full freedom and flexibility
- ✓ Gains on speed and efficiency:
  - Less CPU hours for a fit
  - Usage of new technologies
    - ✓ New hardware
    - ✓ New libraries
  - Usage of Gradient Descent methods
- ✓ Consequences
  - **Speed-up of research:** faster to develop, test, run
  - **More studies available**
  - Example: **fitting the methodology** (hyperparameter scan)



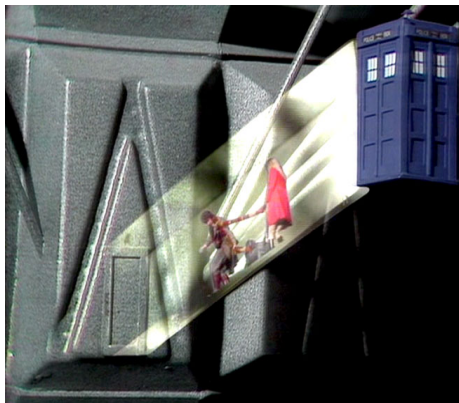
# A new generation of PDF fitting

- ✓ Rationalization of development wrt to older codes
  - Easier and faster development
  - Object orientation: full freedom and flexibility
- ✓ Gains on speed and efficiency:
  - Less CPU hours for a fit
  - Usage of new technologies
    - ✓ New hardware
    - ✓ New libraries
  - Usage of Gradient Descent methods
- ✓ Consequences
  - **Speed-up of research**: faster to develop, test, run
  - **More studies available**
  - Example: **fitting the methodology** (hyperparameter scan)



# The art of the hyperparameter selection

Just as technology has changed the way movies are done, 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



Ground Level Arcadia Breakdown



Ground Level Arcadia Breakdown

2014

# Scan over hyperparameters: fitting the methodology

The main goal of NNPDF was to reduce the bias introduced in the PDF fits by the choice of the functional form of the PDFs, but. . .

- NN are defined by set of parameters
- Humans are good at recognising patterns
- ✗ selecting the right parameters is a slow process and success is not guaranteed



To overcome these issues we implement a **hyperparameter scan**: let the computer decide automatically

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

The next step: fitting the whole methodology

# Scan over hyperparameters: fitting the methodology

The main goal of NNPDF was to reduce the bias introduced in the PDF fits by the choice of the functional form of the PDFs, but. . .

- NN are defined by set of parameters
- Humans are good at recognising patterns
- ✗ selecting the right parameters is a slow process and success is not guaranteed



To overcome these issues we implement a **hyperparameter scan**: let the computer decide automatically

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

The next step: fitting the whole methodology

# Scan over hyperparameters: fitting the methodology

The main goal of NNPDF was to reduce the bias introduced in the PDF fits by the choice of the functional form of the PDFs, but. . .

- NN are defined by set of parameters
- Humans are good at recognising patterns
- ✗ selecting the right parameters is a slow process and success is not guaranteed



To overcome these issues we implement a **hyperparameter scan**: let the computer decide automatically

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

The next step: fitting the whole methodology

# Scan over hyperparameters: fitting the methodology

The main goal of NNPDF was to reduce the bias introduced in the PDF fits by the choice of the functional form of the PDFs, but. . .

- NN are defined by set of parameters
- Humans are good at recognising patterns
- ✗ selecting the right parameters is a slow process and success is not guaranteed



To overcome these issues we implement a **hyperparameter scan**: let the computer decide automatically

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

The next step: fitting the whole methodology



## Scan over hyperparameters: fitting the methodology

The main goal of NNPDF was to reduce the bias introduced in the PDF fits by the choice of the functional form of the PDFs, but. . .

- NN are defined by set of parameters
- Humans are good at recognising patterns
- ✗ selecting the right parameters is a slow process and success is not guaranteed



To overcome these issues we implement a **hyperparameter scan**: let the computer decide automatically

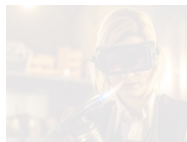
- ✓ Scan over thousands of hyperparameter combinations
- ✓ Define a reward function to grade the model

**The next step: fitting the whole methodology**

# Hyperoptimization: The reward function

The human component is not completely detached it is necessary to define a reward function by choosing the characteristics we find desirable in a fit:

- Goodness of the fit.
- Smoothness of the result.
- Time it takes to complete the full fit.
- Generalization power to future exp data.



Selecting a good reward function (although can be highly non-trivial) offers several advantages:

- ✓ Several fits can present similar goodness but differ in other features.
- ✓ Reduced bias on parameter selection.
- ✓ Automatic scan of the whole space of parameters.

# Hyperoptimization: The reward function

The human component is not completely detached it is necessary to define a reward function by choosing the characteristics we find desirable in a fit:

- Goodness of the fit.
- Smoothness of the result.
- Time it takes to complete the full fit.
- Generalization power to future exp data.



Selecting a good reward function (although can be highly non-trivial) offers several advantages:

- ✓ Several fits can present similar goodness but differ in other features.
- ✓ Reduced bias on parameter selection.
- ✓ Automatic scan of the whole space of parameters.

## Hyperoptimization: The reward function

The human component is not completely detached it is necessary to define a reward function by choosing the characteristics we find desirable in a fit:

- Goodness of the fit.
- Smoothness of the result.
- Time it takes to complete the full fit.
- Generalization power to future exp data.



Selecting a good reward function (although can be highly non-trivial) offers several advantages:

- ✓ Several fits can present similar goodness but differ in other features.
- ✓ Reduced bias on parameter selection.
- ✓ Automatic scan of the whole space of parameters.

# Hyperoptimization: The reward function

The human component is not completely detached it is necessary to define a reward function by choosing the characteristics we find desirable in a fit:

- **Goodness of the fit.**
- Smoothness of the result.
- Time it takes to complete the full fit.
- **Generalization power to future exp data.**



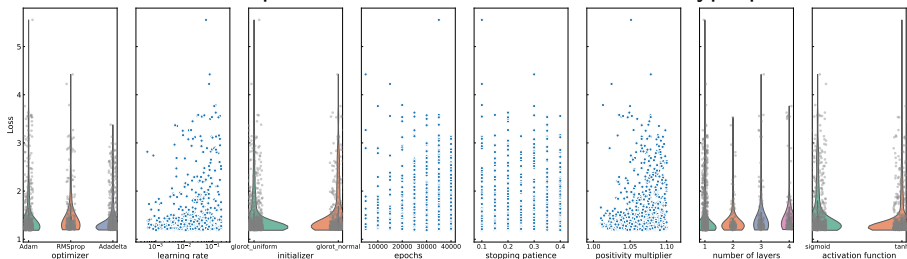
Example of function to hyperoptimize:

$$\text{Loss} = \frac{1}{2} (\chi_{\text{fit}}^2 + \chi_{\text{generalization set}}^2)$$

Where “generalization set” corresponds to experimental data that did not enter the fit.

# 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

## Comparison between new and old methodologies

n3fit is fully implemented now and produces results which are compatible with previous releases of NNPDF at a lesser cost.

As a proof of concept we present a fit done with n3fit after a run of the automated hyperoptimization

	n3fit	NNPDF 3.1
$\chi^2$	1.149	1.158
Avg time	70 minutes	35 hours
Good replicas	95%	70%

Note: Good replicas refer to those which produce a good fit in the allotted number of iterations.

- Same dataset selection
- Same positivity constraints
- ✓ Very different methodologies
- ✓ Very similar fit goodness
- ✓ Orders of magnitude faster

## Comparison between new and old methodologies

n3fit is fully implemented now and produces results which are compatible with previous releases of NNPDF at a lesser cost.

As a proof of concept we present a fit done with n3fit after a run of the automated hyperoptimization

	n3fit	NNPDF 3.1
$\chi^2$	1.149	1.158
Avg time	70 minutes	35 hours
Good replicas	95%	70%

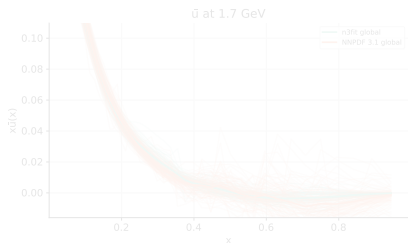
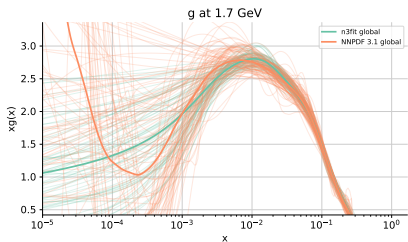
Note: Good replicas refer to those which produce a good fit in the allotted number of iterations.

- Same dataset selection
- Same positivity constraints
- ✓ Very different methodologies
- ✓ Very similar fit goodness
- ✓ Orders of magnitude faster



# Result comparison: replica-by-replica

Comparison. with the same selection of data, of the old and new codes.



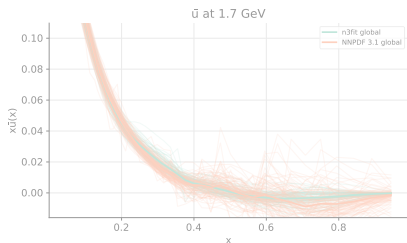
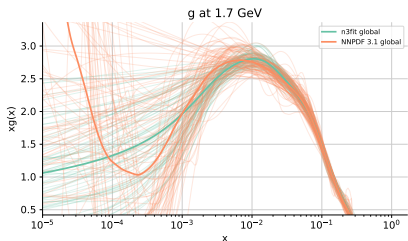
- ✓ Smoother results in the data region
- ✓ More replicas satisfy post-fit requirements
- ✓ Compatibility of old and new results

Which translates to

- ✓ Many more studies can be performed at the same cost  
→ more science
- ✓ Leading to a more accurate PDF determination

# Result comparison: replica-by-replica

Comparison. with the same selection of data, of the old and new codes.



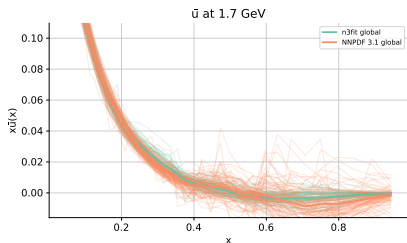
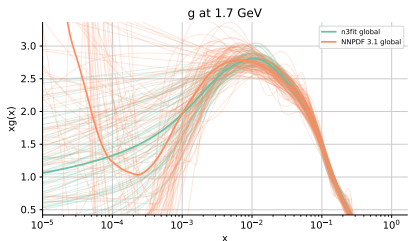
- ✓ Smoother results in the data region
- ✓ More replicas satisfy post-fit requirements
- ✓ Compatibility of old and new results

Which translates to

- ✓ Many more studies can be performed at the same cost  
→ more science
- ✓ Leading to a more accurate PDF determination

# Result comparison: replica-by-replica

Comparison. with the same selection of data, of the old and new codes.



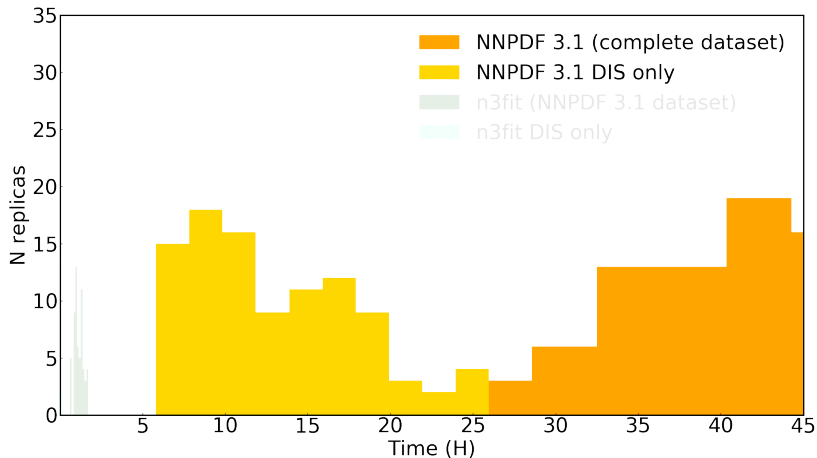
- ✓ Smoother results in the data region
- ✓ More replicas satisfy post-fit requirements
- ✓ Compatibility of old and new results

Which translates to

- ✓ Many more studies can be performed at the same cost  
→ more science
- ✓ Leading to a more accurate PDF determination

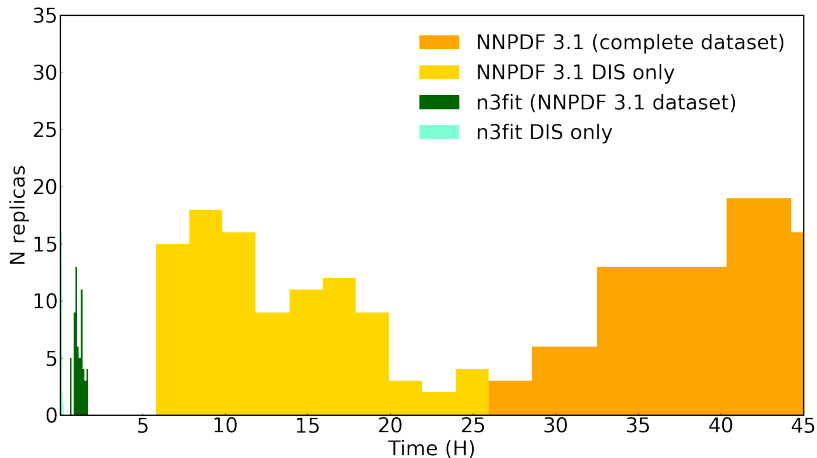
## Practical example: time distribution of replica fitting

An important outcome of the development of `n3fit` is that the performance of the fits has been greatly increased, enabling new studies.

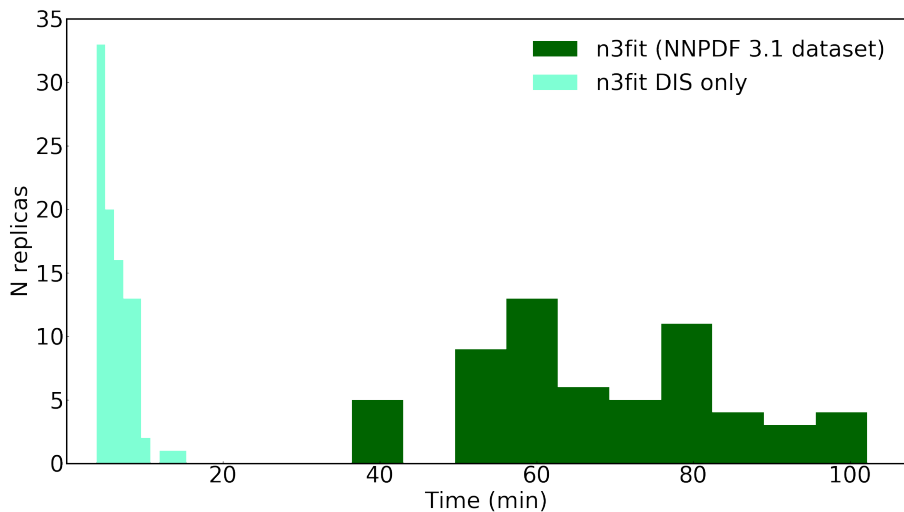


## Practical example: time distribution of replica fitting

An important outcome of the development of `n3fit` is that the performance of the fits has been greatly increased, enabling new studies.



# From hours to minutes



# Summary

- ✓ **Towards NNPDF 4.0:** NNPDF machinery for PDF fitting is now more powerful, faster and flexible.
  - ✓ Faster run times: iterate over different choices of models or parameters.
  - ✓ Smoother results, less outliers.
  - ✓ Novel research now accessible.
  - ✓ The framework allows full customization *by design*.
- **The cost of doing new studies is reduced, both the development/implementation and the raw computational cost.**

# Thanks!



## Differences with respect to the old methodology

### NNPDF 3.1 code

- Genetic Algorithm optimizer
- One network per flavour
- Sum rules imposed outside of optimization
- Preprocessing fixed per each of the replicas
- C++ monolithic codebase
- Fit parameters manually chosen (i.e., manual optimization of hyperparameter)
- In-house ML framework

### n3fit code

- Gradient Descent optimization
- One network for all flavours
- Sum rules imposed during optimization
- Preprocessing fitted within replicas
- Python object oriented codebase
- Fit parameters chosen automatically (hyperparameter scan)
- Complete freedom for choosing the ML library: i.e., tensorflow.

## Differences with respect to the old methodology

### NNPDF 3.1 code

- **Genetic Algorithm optimizer**
- One network per flavour
- Sum rules imposed outside of optimization
- Preprocessing fixed per each of the replicas
- C++ monolithic codebase
- **Fit parameters manually chosen (i.e., manual optimization of hyperparameter)**
- In-house ML framework

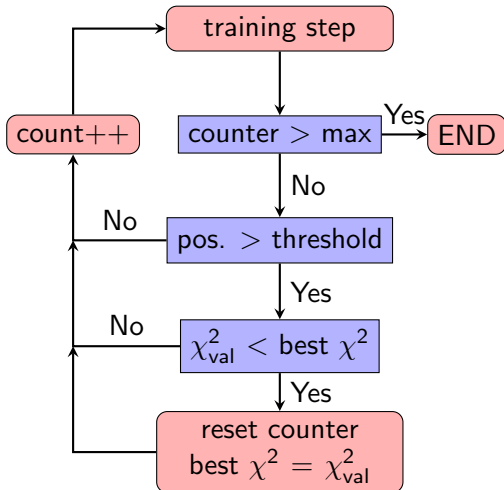
### n3fit code

- **Gradient Descent optimization**
- One network for all flavours
- Sum rules imposed during optimization
- Preprocessing fitted within replicas
- Python object oriented codebase
- **Fit parameters chosen automatically (hyperparameter scan)**
- Complete freedom for choosing the ML library: i.e., tensorflow.

# Stopping

Stopping method:

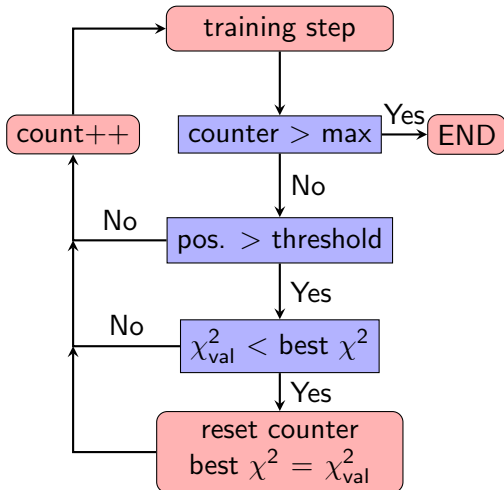
**Look-back method where positivity passes**



# Stopping

Stopping method: **Look-back method where positivity passes**

Early stopping: reduce overfitting

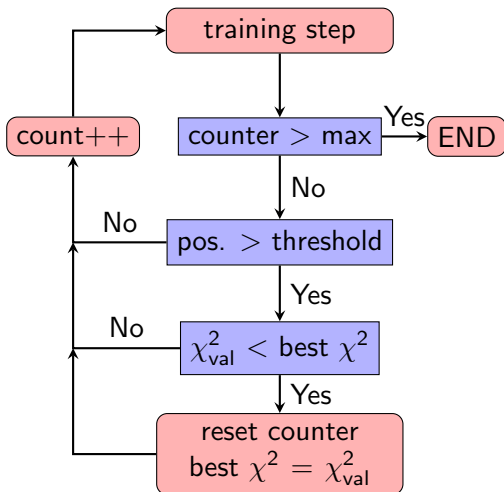


# Stopping

Stopping method: **Look-back method where positivity passes**

Early stopping: reduce overfitting

- ✓ Minimize  $\chi^2$  of validation sets



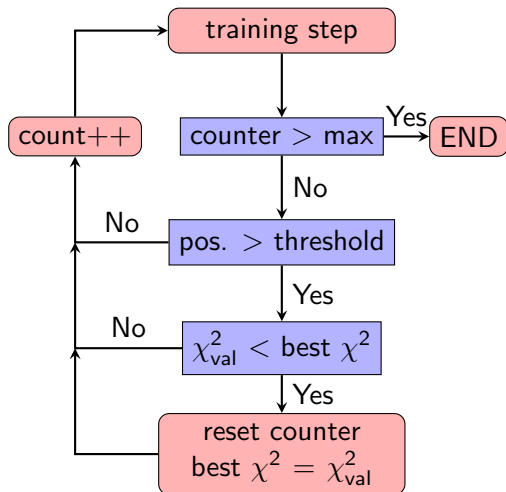
# Stopping

Stopping method: **Look-back method where positivity passes**

Early stopping: reduce overfitting

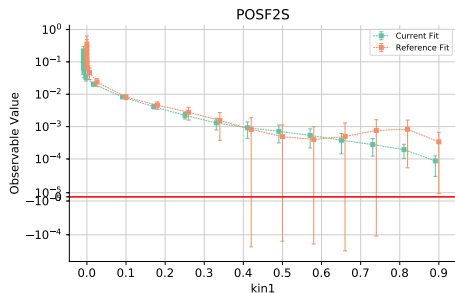
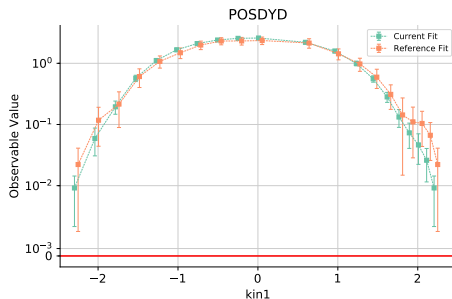
✓ Minimize  $\chi^2$  of validation sets

✓ Enforces positivity constraints

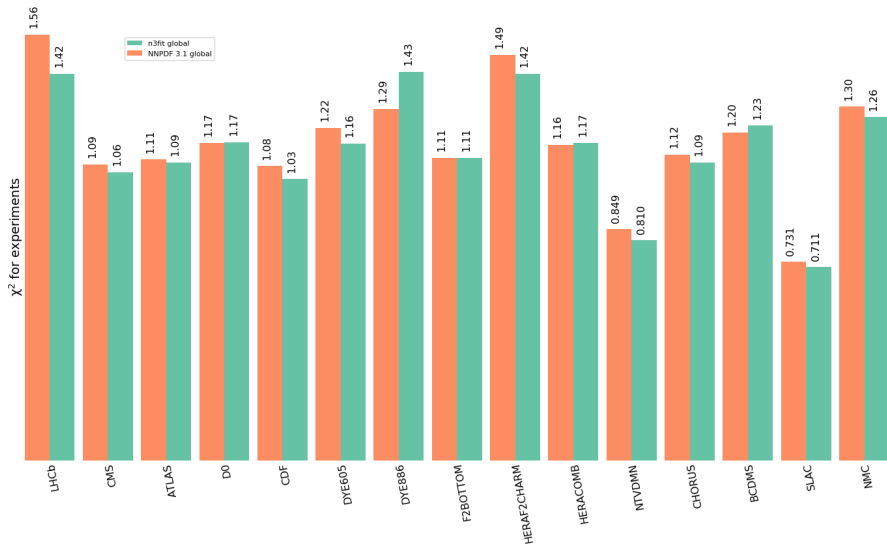


# Positivity constrained

Once all these considerations are applied, we obtain no replicas of negative positivity.

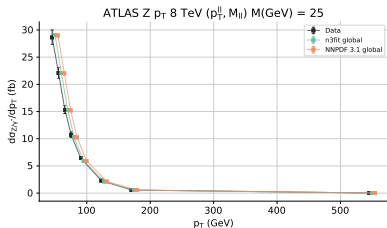


# Per-experiment results



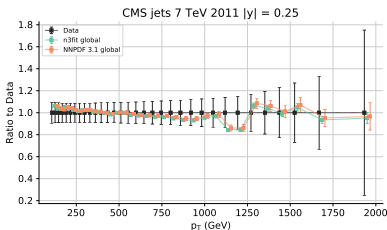


# Comparison to data



→ Results compatible with NNPDF 3.1

→ Not only a similar  $\chi^2$ -goodness but also similar per-point results



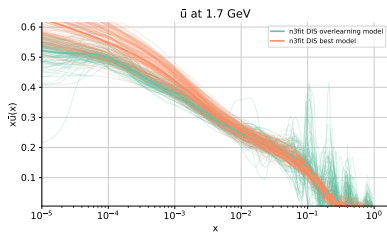
✓ The new methodology is compatible with the previous one!

## Warning: overfitting!

*With great power comes great responsibility.*

An unsupervised parameter scan is dangerous: it can find that overfitting is preferable.

- ✗ It did minimise the validation!
- ✗ Hyperopt is able to trick cross-validation when choosing the model.



## Warning: overfitting!

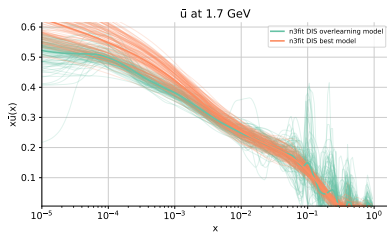
*With great power comes great responsibility.*

An unsupervised parameter scan is dangerous: it can find that overfitting is preferable.

- ✗ It did minimise the validation!
- ✗ Hyperopt is able to trick cross-validation when choosing the model.

Solution:

- ✓ Create a test-set:  
Take a few experiments out of the hyperparameter scan and use them to probe the generalization power of the network



# The test set

The creation of a properly defined test set is quite a convoluted task. For [hep-ph/1907.05075] we have restricted ourself to the following two items:

- Redundant datasets: we select processes with more than one dataset of experimental data..
- Smaller kinetic range: of the redundant datasets we select the one that covers a smaller kinematic range (in practice, we take out the one whose  $x_{\min}$  is bigger).

Finally the hyperoptimization itself is performed on a combination of the validation loss of each fit and the  $\chi^2$  of the fit to the testing set. Furthermore the fits are tested for stability in order to remove potentially