Methodological improvements in PDF determination

Juan M Cruz-Martinez

in collaboration with: S. Carrazza, J. Urtasun-Elizari, E. Villa

hep-ph/1907.05075

N3PDF

Machine Learning • PDFs • QCD

James Stirling Memorial Conference, Durham (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.

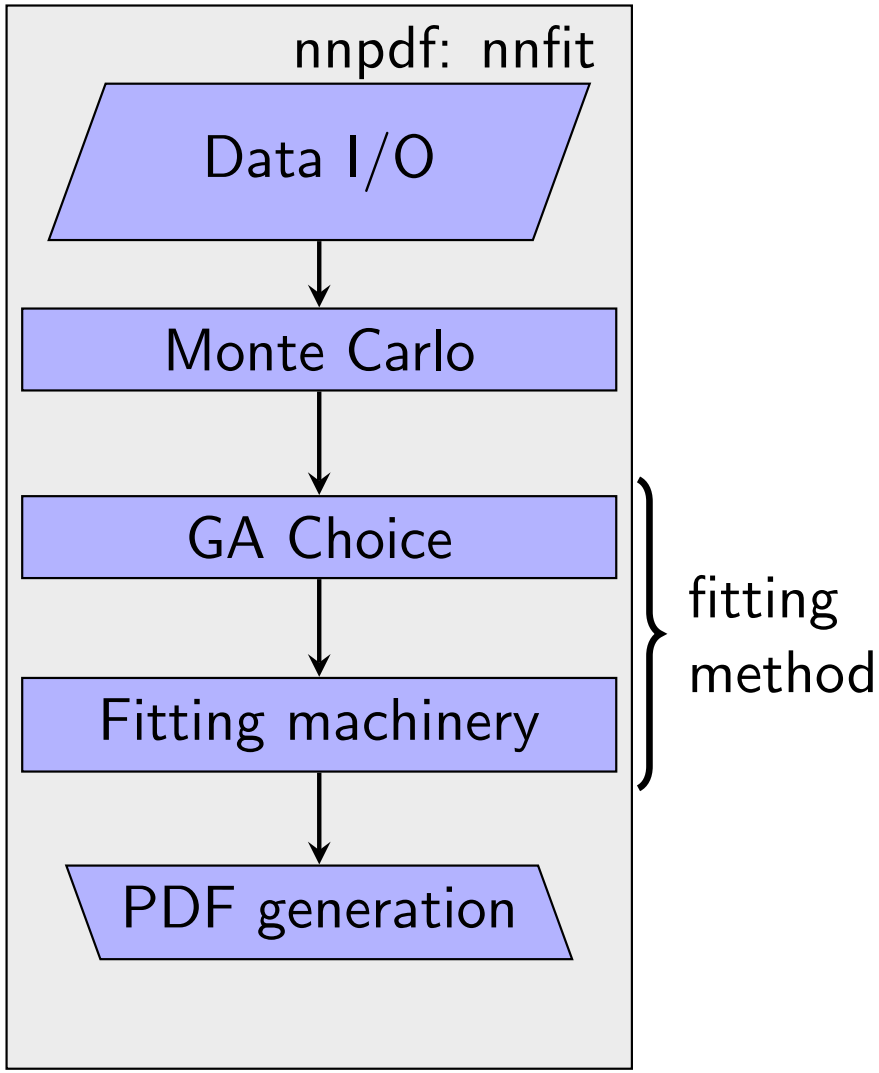
Outline

1. A new methodology, codename `n3fit`
   - Motivation: speed & flexibility $\rightarrow$ more physics
   - Design choices

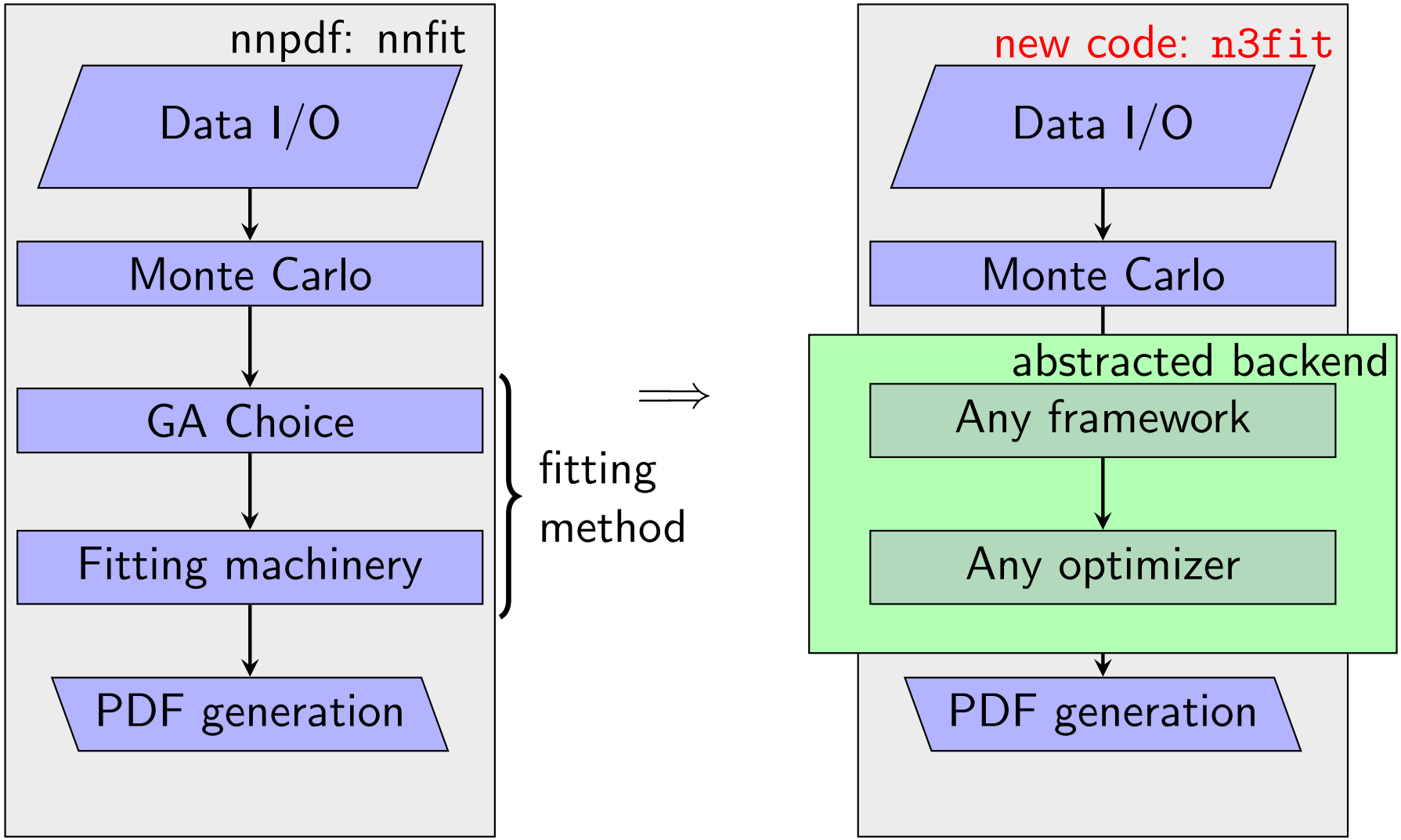
2. Codename `n3fit`
   - In detail
   - Hyperoptimization: fitting the methodology
   - Result examples

3. Accelerating the fit
   - Handcrafting operations
   - Hardware acceleration

The goal: towards new methodologies

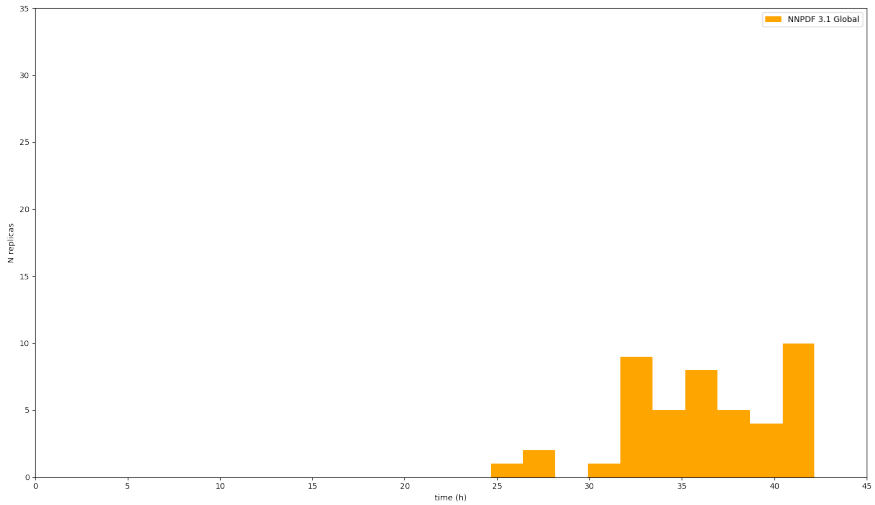


The goal: towards new methodologies



Motivation: more studies available

✓ Rationalization of development
- Easier and faster development
- OOP: full freedom and flexibility

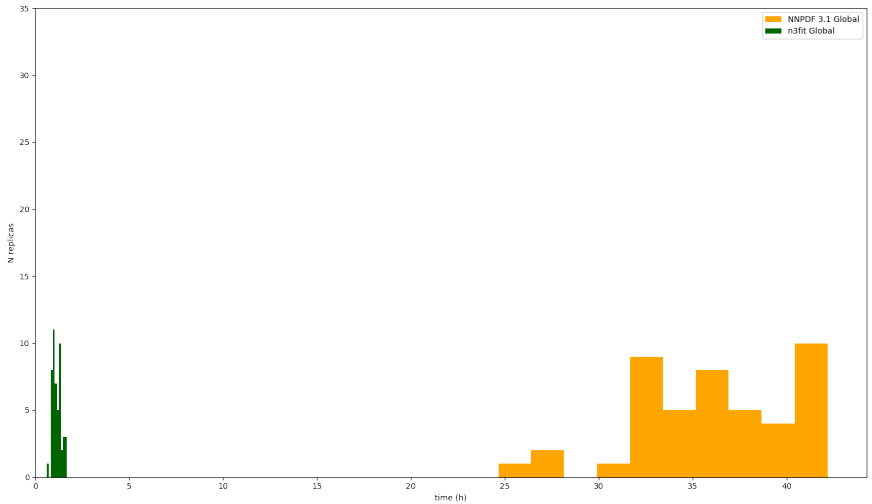


✓ Gains on speed and efficiency:
- Less CPU hours for a fit
- Usage of new technologies
  - ✓ GPU/FPGA
  - ✓ ML libraries

✓ Consequences
- **Speed-up of research**
- **More studies available**
- **Example: fitting the methodology**

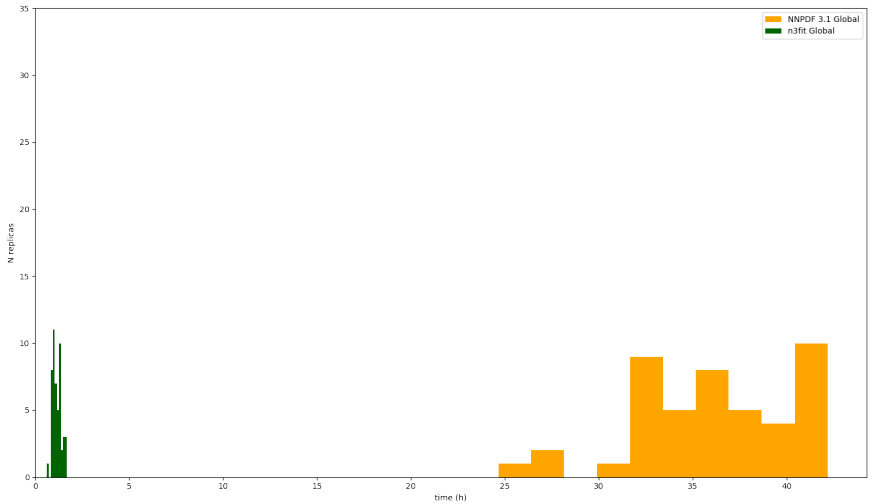
Motivation: more studies available

- ✓ Rationalization of development
  - Easier and faster development
  - OOP: full freedom and flexibility
- ✓ Gains on speed and efficiency:
  - Less CPU hours for a fit
  - Usage of new technologies
    - ✓ GPU/FPGA
    - ✓ ML libraries
- ✓ Consequences
  - Speed-up of research
  - More studies available
  - Example: fitting the methodology



Motivation: more studies available

- ✓ Rationalization of development
  - Easier and faster development
  - OOP: full freedom and flexibility
- ✓ Gains on speed and efficiency:
  - Less CPU hours for a fit
  - Usage of new technologies
    - ✓ GPU/FPGA
    - ✓ ML libraries
- ✓ Consequences
  - **Speed-up of research**
  - **More studies available**
  - **Example: fitting the methodology**



Language and framework

```
Language:  python
      > version:  3.7
```

- ✓ Widely used and easy to learn
- ✓ Supported by most ML libraries:
  (Tensorfow, Pytorch, CNTK, ...)

```
Framework:  Keras
      > backend:  Tensorflow
```

- ✓ High level of abstraction
- ✓ Powerful features of Tensorflow
- ✓ Trivially change between (supported) libraries

- ✓ CPU parallelization
- ✓ GPU parallelization
- ✓ FPGA support

Language and framework

`Language: python`

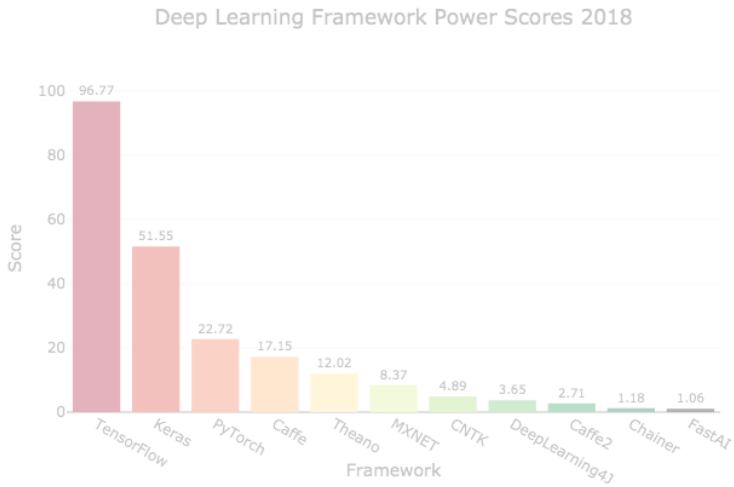
`> version: 3.7`

- ✓ Widely used and easy to learn
- ✓ Supported by most ML libraries: (Tensorfow, Pytorch, CNTK, . . .)

`Framework: Keras`

`> backend: Tensorflow`

- ✓ High level of abstraction
- ✓ Powerful features of Tensorflow
- ✓ Trivially change between (supported) libraries



- ✓ CPU parallelization
- ✓ GPU parallelization
- ✓ FPGA support

Language and framework

`Language: python`

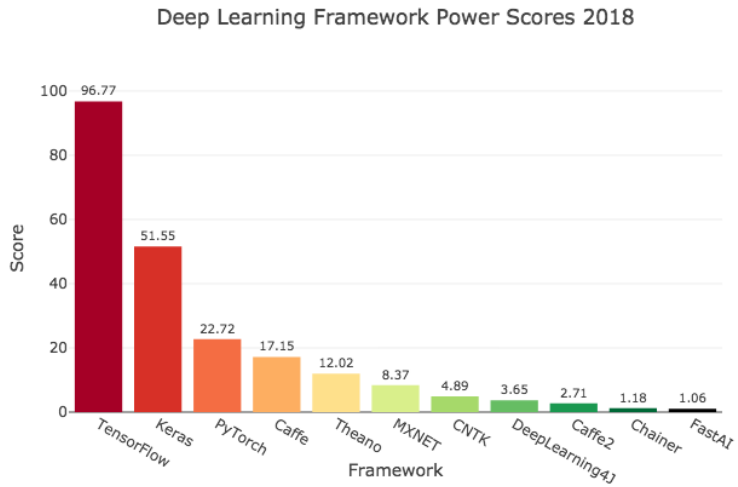
`> version: 3.7`

- ✓ Widely used and easy to learn
- ✓ Supported by most ML libraries: (Tensorfow, Pytorch, CNTK, ...)

`Framework: Keras`

`> backend: Tensorflow`

- ✓ High level of abstraction
- ✓ Powerful features of Tensorflow
- ✓ Trivially change between (supported) libraries



Plot Source

- ✓ CPU parallelization
- ✓ GPU parallelization
- ✓ FPGA support

Note: the metric in the figure includes GitHub activity and ArXiv articles: Research and development.

The Loss Function

The fitting strategy is based on the minimization of the χ^2,

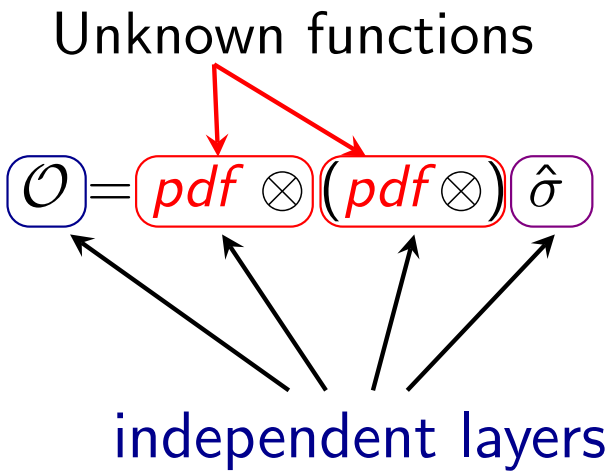
$$\chi^2 = \frac{1}{N} \sum \left(\mathcal{O}^i - \mathcal{D}^i\right) \sigma_{ij}^{-1} \left(\mathcal{O}^j - \mathcal{D}^j\right)$$

N: number of data points

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

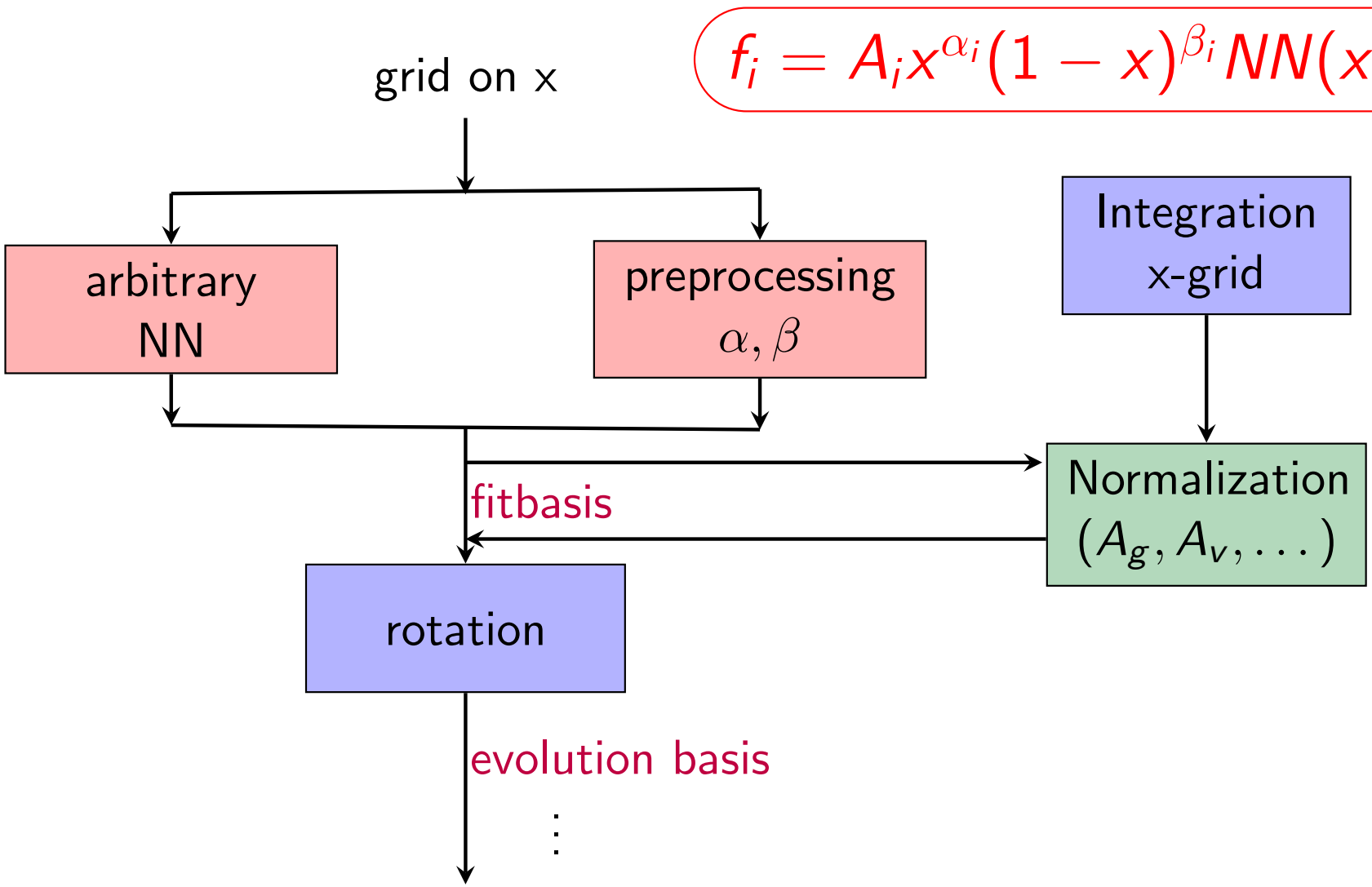
$\mathcal{D}^i$: experimental data point

σ_{ij}: covariance matrix

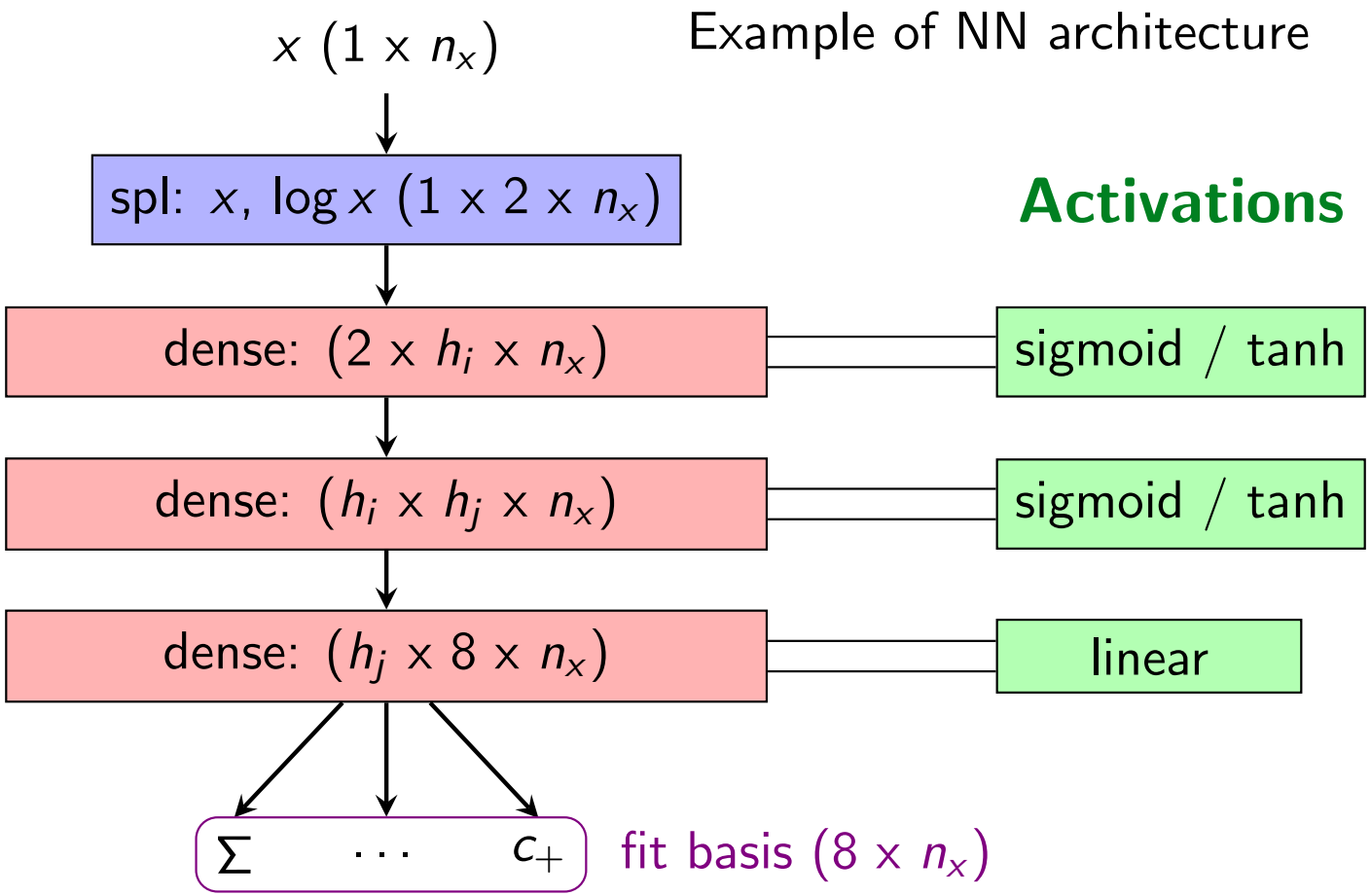


Note: The partonic cross section σ correspond to APFELgrid tables as described in hep-ph/1605.02070

The PDF model

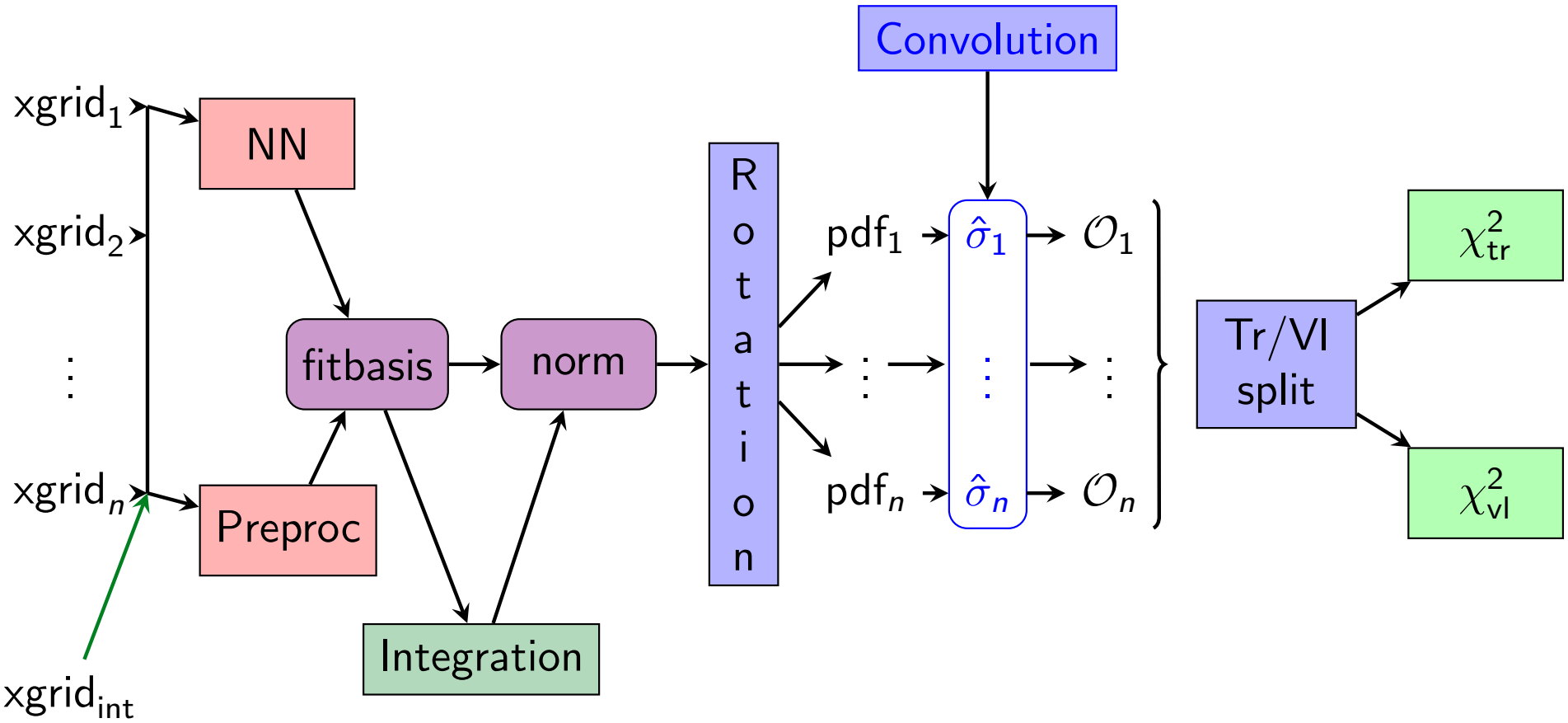


The Neural Network: $NN(x)$



Swapping and testing different network architectures is a matter of seconds: we can systematically scan and find the best model.

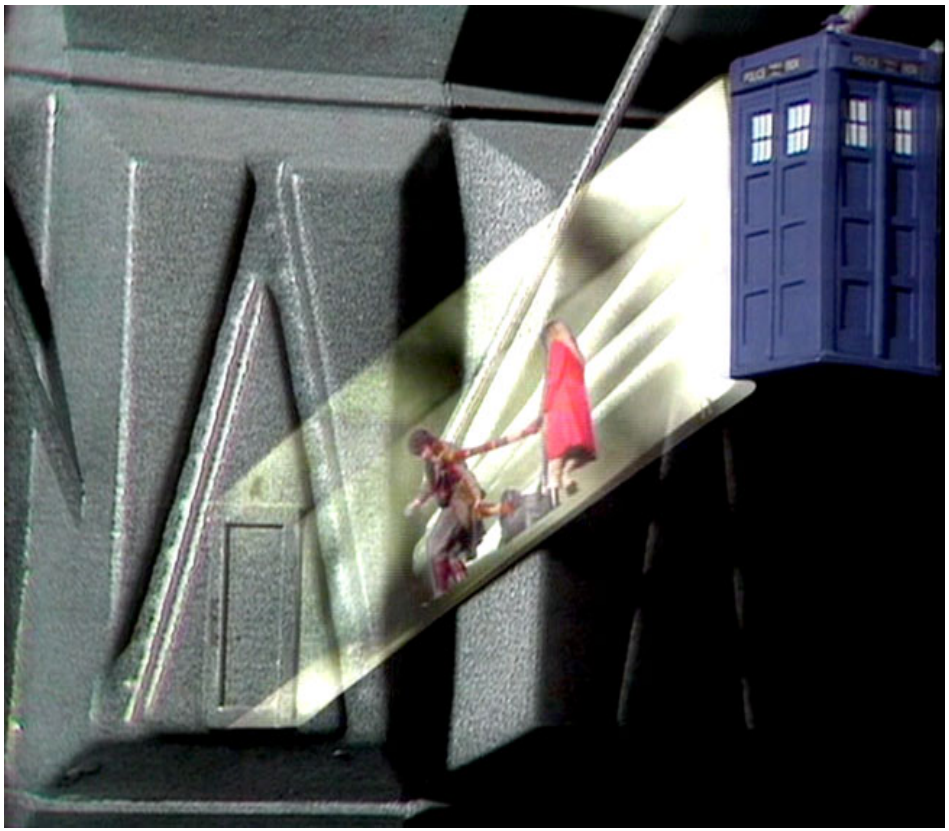
The full model



The art of the hyperparameter selection

With technology, what used to be hand-made can now be computer generated.

1978



2014

We can now let `n3fit` select its own hyperparameters

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...

$\longrightarrow$ NN are defined by set of parameters

$\longrightarrow$ Not clear which is the best choice

$\longrightarrow$ Humans are usually good at recognising patterns

✗ but the wiser decision is not guaranteed

In order to overcome these issues we implement a hyperparameter scan: let the computer decide

✓ 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

→ Not clear which is the best choice

→ Humans are usually good at recognising patterns

✗ but the wiser decision is not guaranteed

In order to overcome these issues we implement a hyperparameter scan: let the computer decide

✓ 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

→ Not clear which is the best choice

→ Humans are usually good at recognising patterns

✗ but the wiser decision is not guaranteed

In order to overcome these issues we implement a hyperparameter scan: let the computer decide

✓ 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

→ Not clear which is the best choice

→ Humans are usually good at recognising patterns

✗ but the wiser decision is not guaranteed

In order to overcome these issues we implement a hyperparameter scan: let the computer decide

✓ 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

→ Not clear which is the best choice

→ Humans are usually good at recognising patterns

✗ but the wiser decision is not guaranteed

In order to overcome these issues we implement a hyperparameter scan: let the computer decide

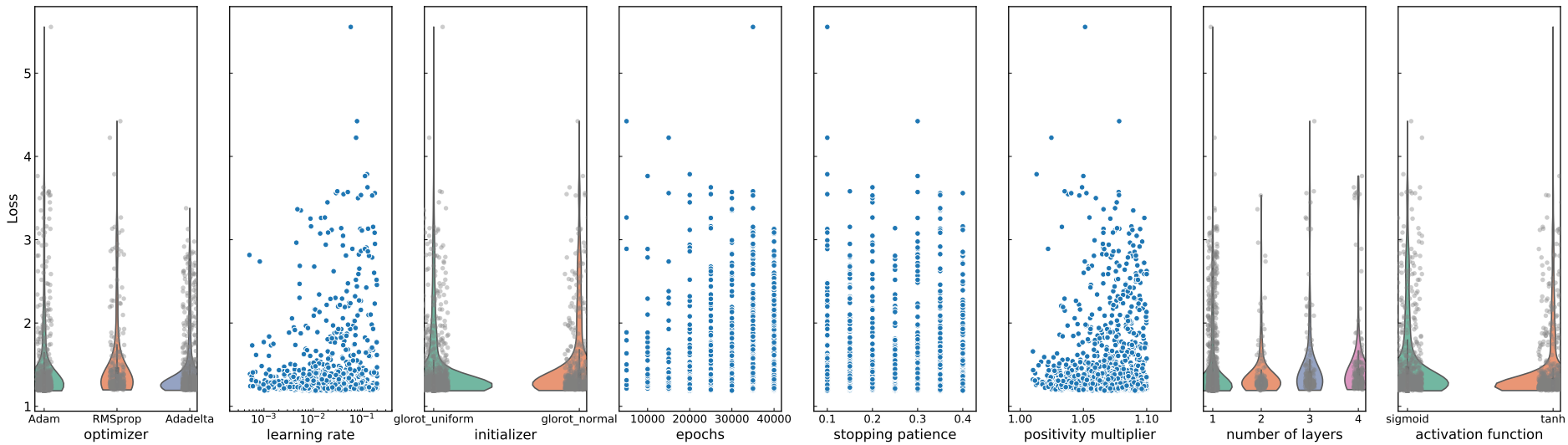
✓ Scan over thousands of hyperparameter combinations

✓ Define a reward function to grade the model

The next step: fitting the whole methodology

Hyperparameter scan

Each blue dot corresponds to a different set of hyperparameters.

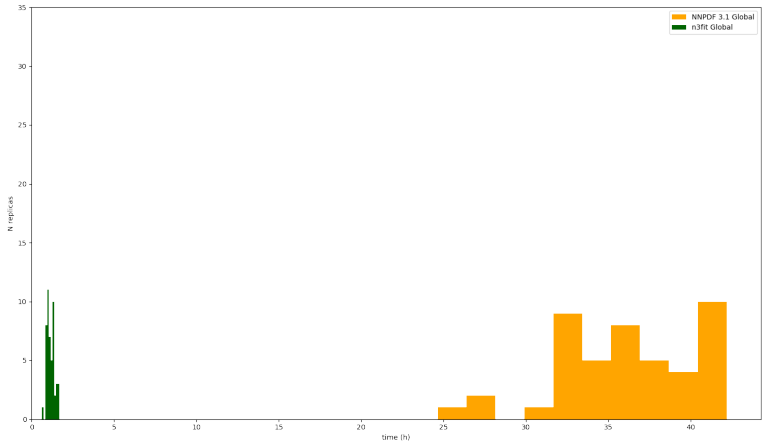


- ✓ Optimizer
- ✓ Initializer
- ✓ Stopping Patience
- ✓ Number of Layers
- ✓ Learning Rate
- ✓ Epochs
- ✓ Positivity Multiplier
- ✓ Activation Function

Results report

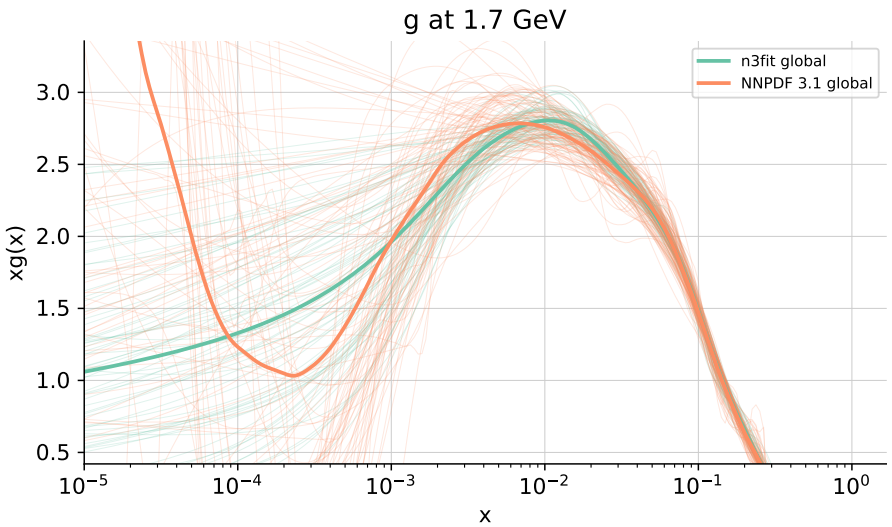
n3fit is fully implemented now and produces results which are compatible with previous releases of NNPDF.

	n3fit	NNPDF 3.1
χ^2	1.149	1.158
Avg time	70 minutes	35 hours
Memory	16 Gb	5 Gb
Stability	95%	70%



As an example we show a comparison using as a baseline a global NNPDF 3.1 NNLO fit against a model created by the hyperoptimization procedure.

Stability



✓ Better stability replica-by-replica

✓ More replicas satisfy post-fit requirements

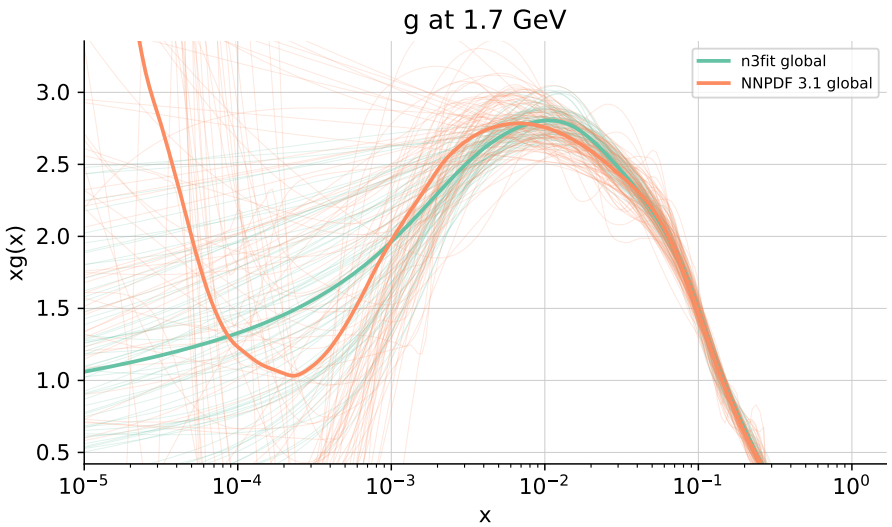
Which translates to

✓ Even smaller computing times!

✓ More complete statistical analysis at the same cost

✓✓ More accurate PDF determination!

Stability



✓ Better stability replica-by-replica

✓ More replicas satisfy post-fit requirements

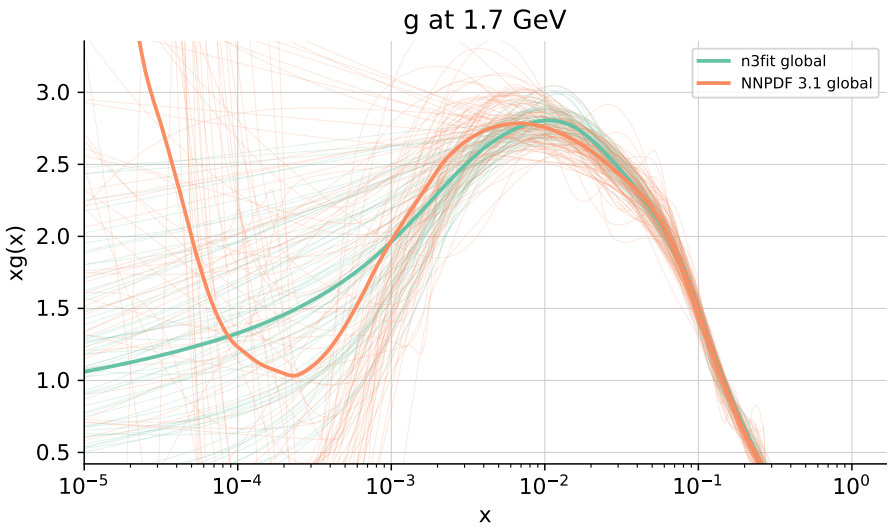
Which translates to

✓ Even smaller computing times!

✓ More complete statistical analysis at the same cost

✓✓ More accurate PDF determination!

Stability

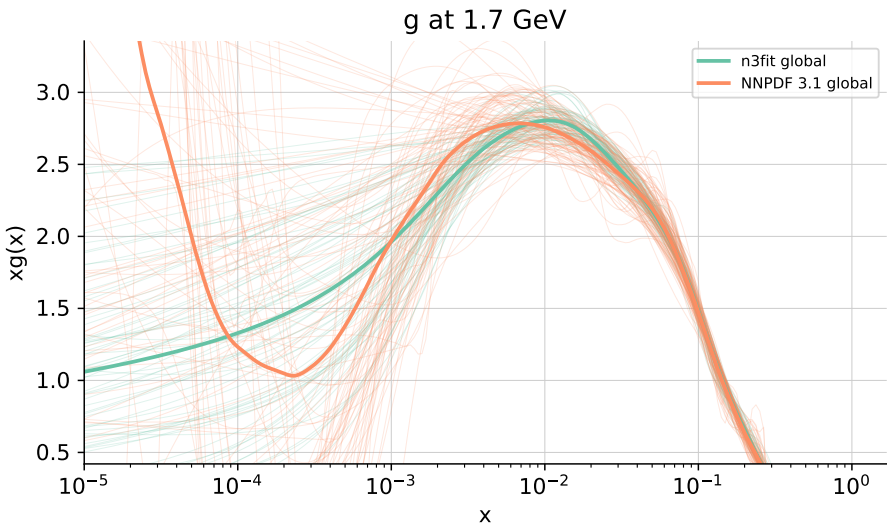


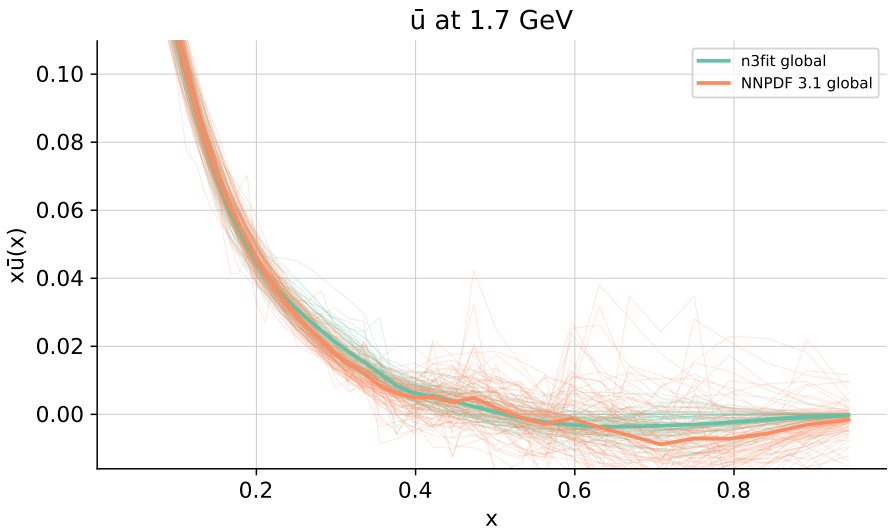
- ✓ Better stability replica-by-replica
- ✓ More replicas satisfy post-fit requirements

Which translates to

- ✓ Even smaller computing times!
- ✓ More complete statistical analysis at the same cost
- ✓✓ More accurate PDF determination!

Stability



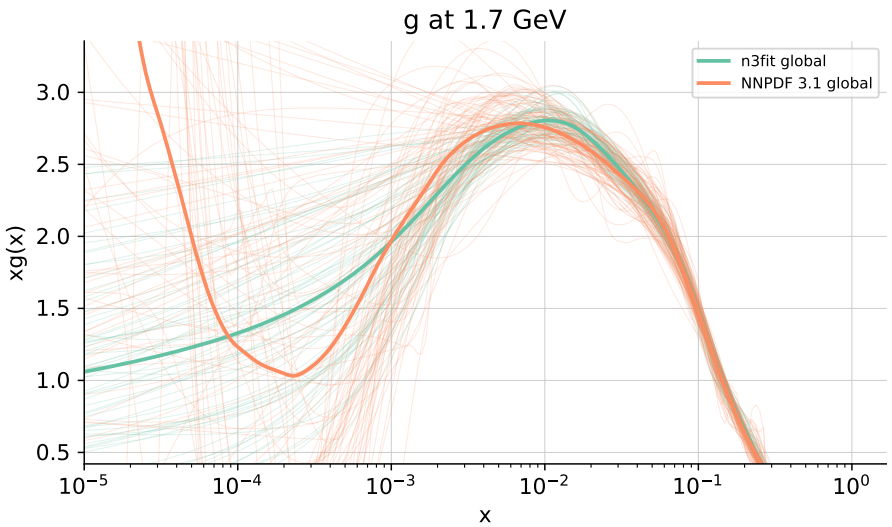


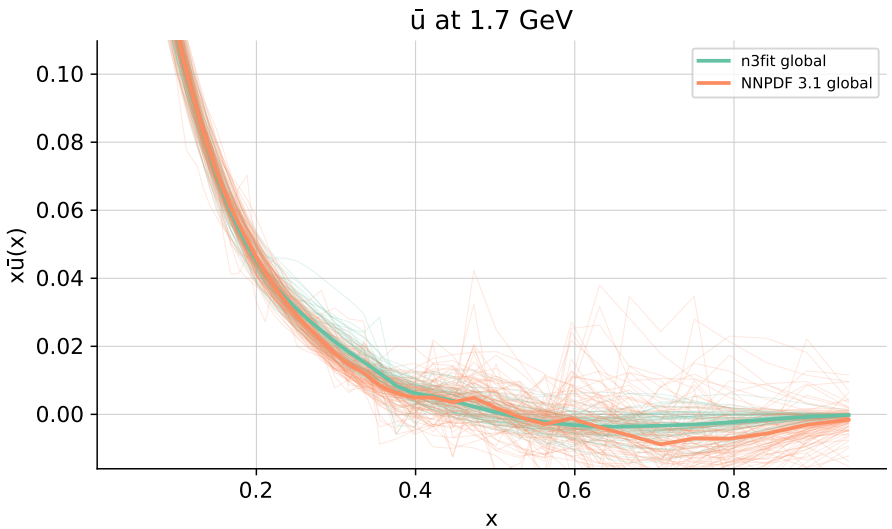
- ✓ Better stability replica-by-replica
- ✓ More replicas satisfy post-fit requirements

Which translates to

- ✓ Even smaller computing times!
- ✓ More complete statistical analysis at the same cost
- ✓✓ More accurate PDF determination!

Stability



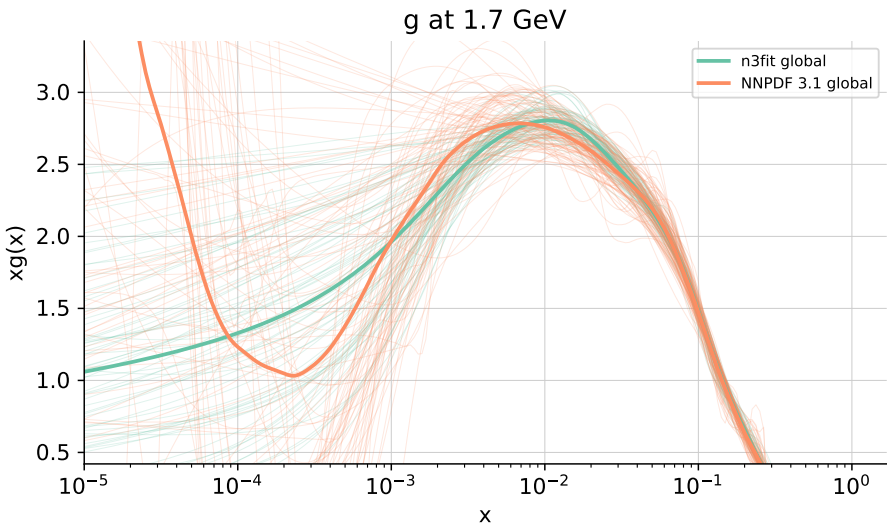


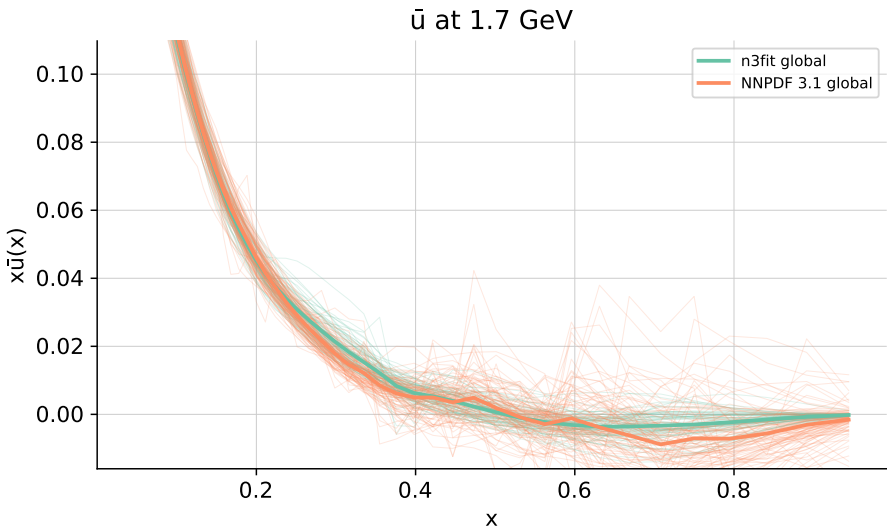
- ✓ Better stability replica-by-replica
- ✓ More replicas satisfy post-fit requirements

Which translates to

- ✓ Even smaller computing times!
- ✓ More complete statistical analysis at the same cost
- ✓✓ More accurate PDF determination!

Stability



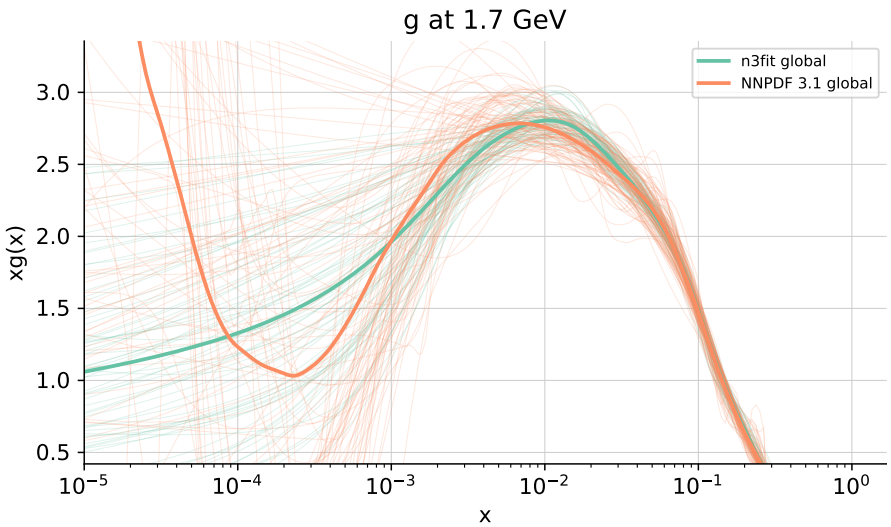


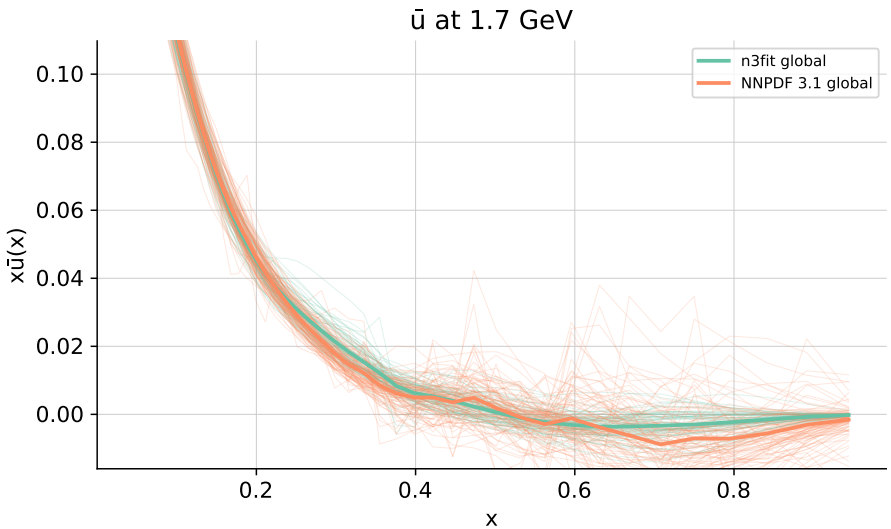
- ✓ Better stability replica-by-replica
- ✓ More replicas satisfy post-fit requirements

Which translates to

- ✓ Even smaller computing times!
- ✓ More complete statistical analysis at the same cost
- ✓✓ More accurate PDF determination!

Stability





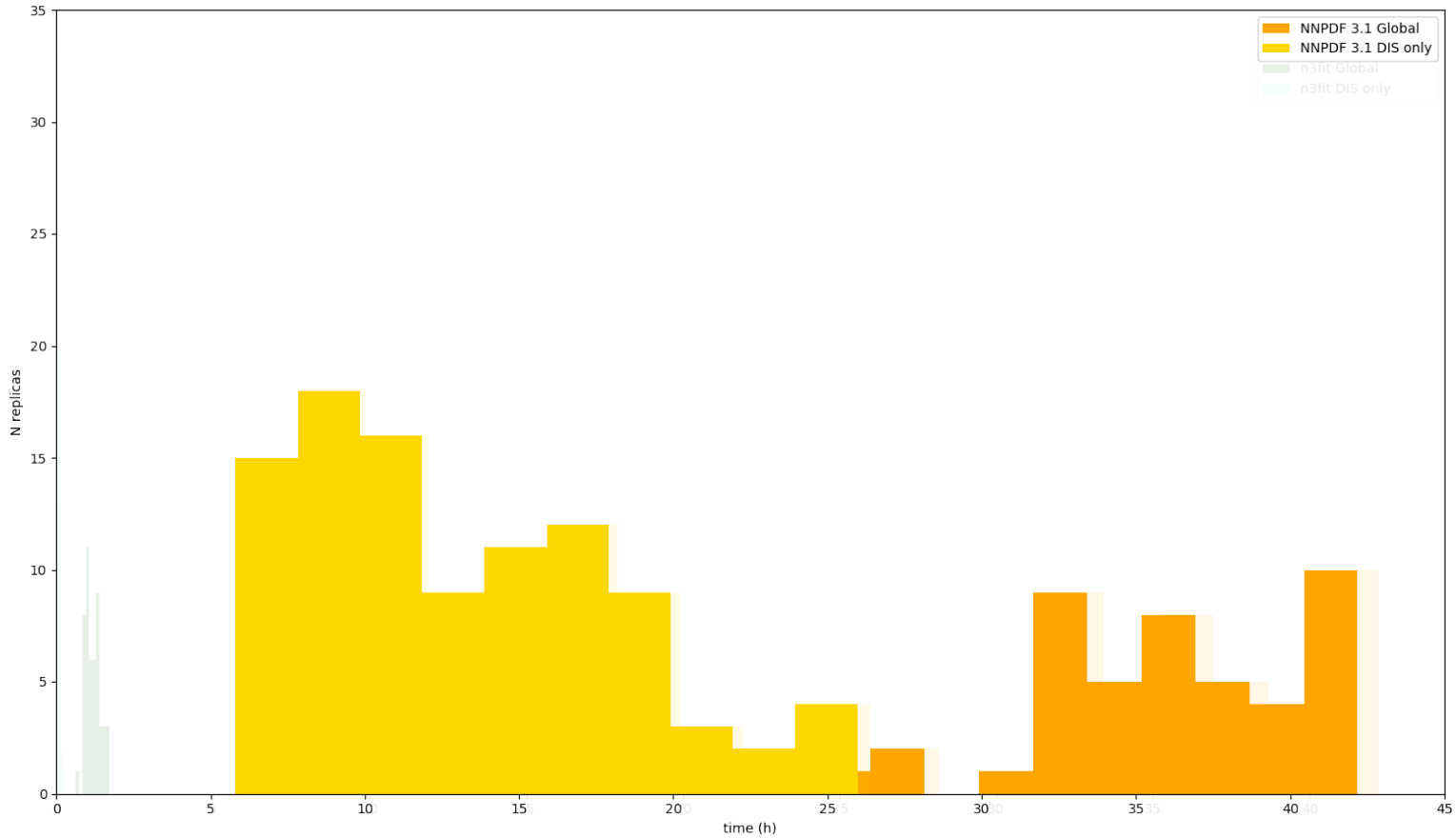
- ✓ Better stability replica-by-replica
- ✓ More replicas satisfy post-fit requirements

Which translates to

- ✓ Even smaller computing times!
- ✓ More complete statistical analysis at the same cost
- ✓✓ More accurate PDF determination!

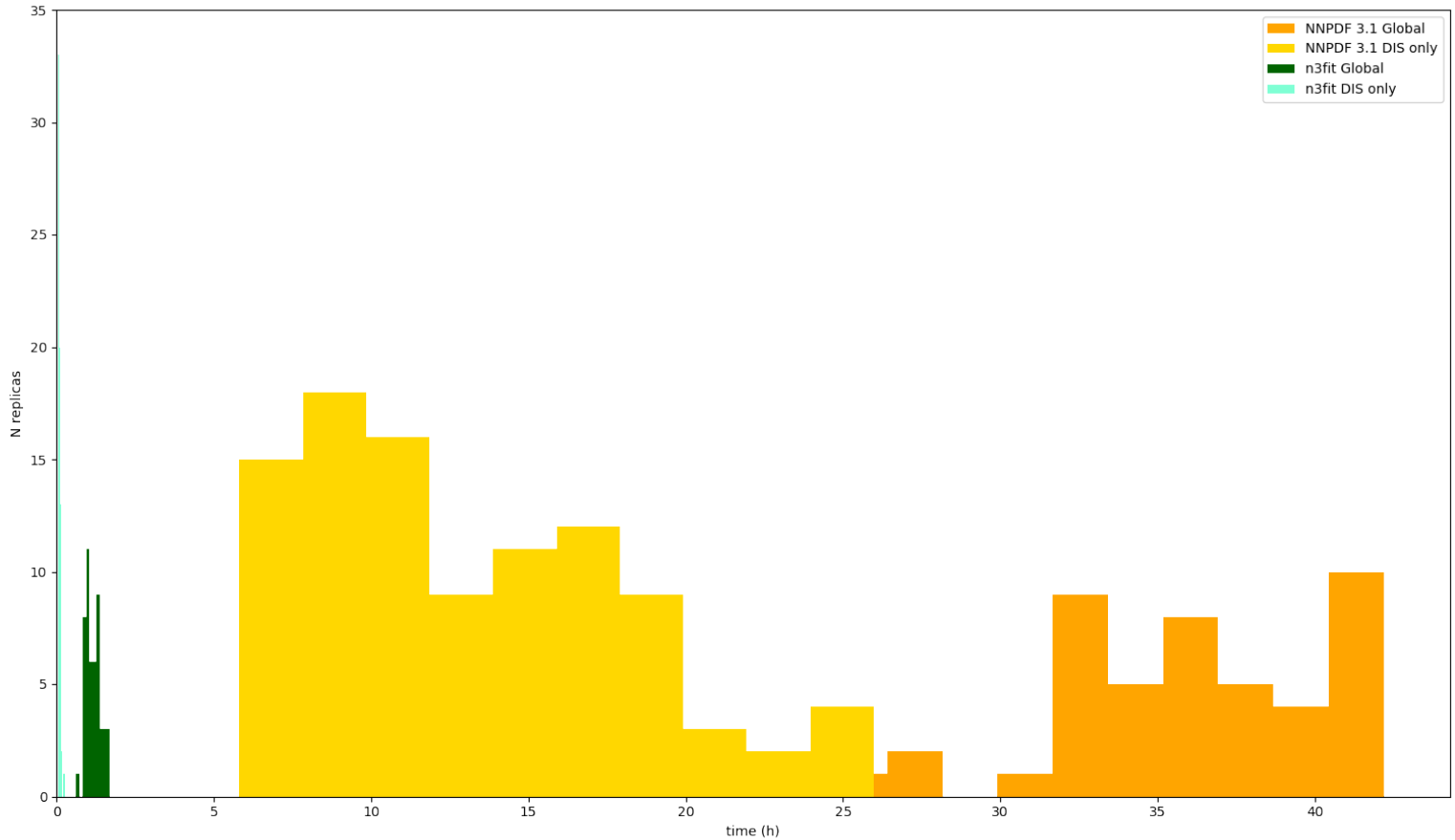
Fit duration distribution

One of the most obvious improvements
has been the performance of the fit, with times of about an hour per replica.



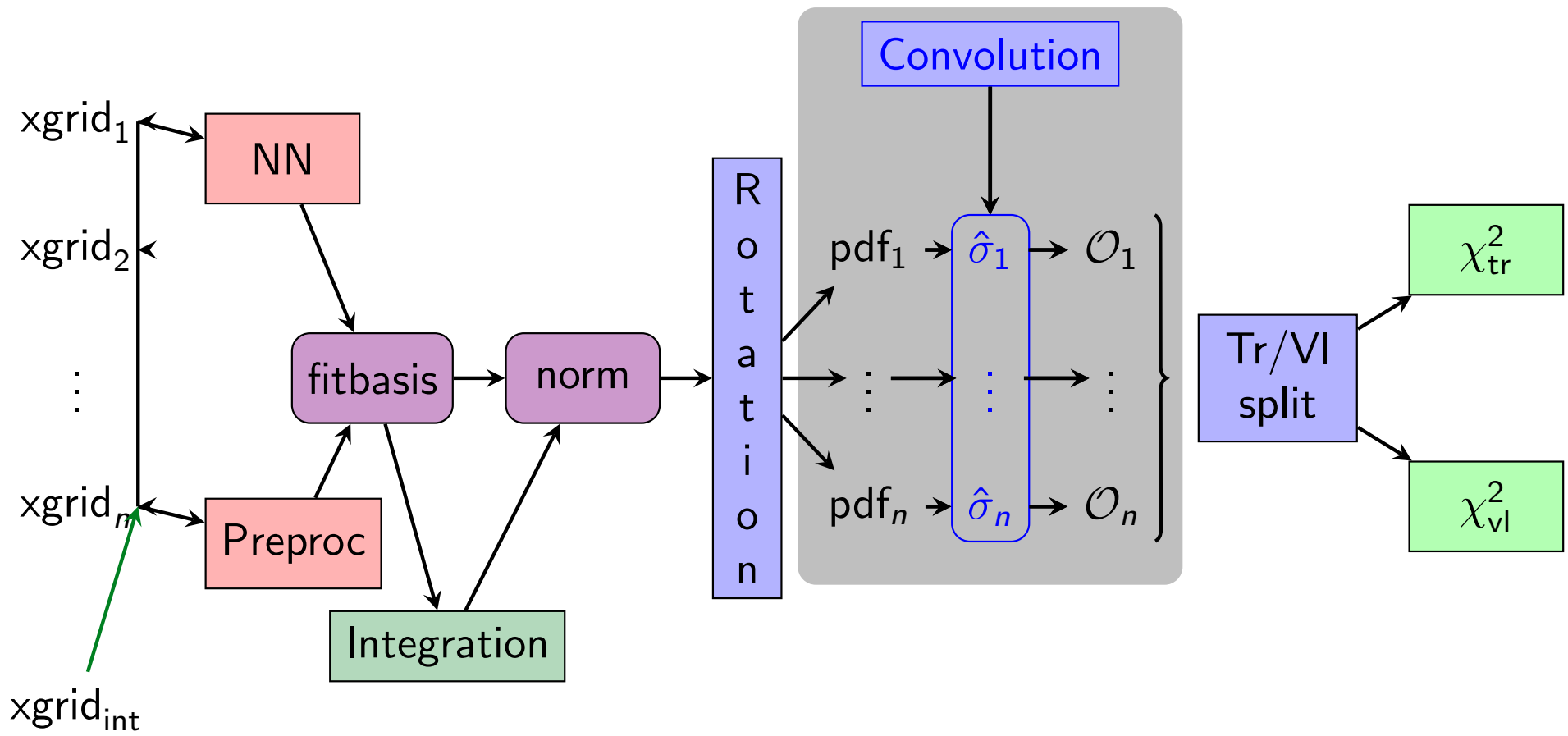
Fit duration distribution

One of the most obvious improvements
has been the performance of the fit, with times of about an hour per replica.



Customizing the operators

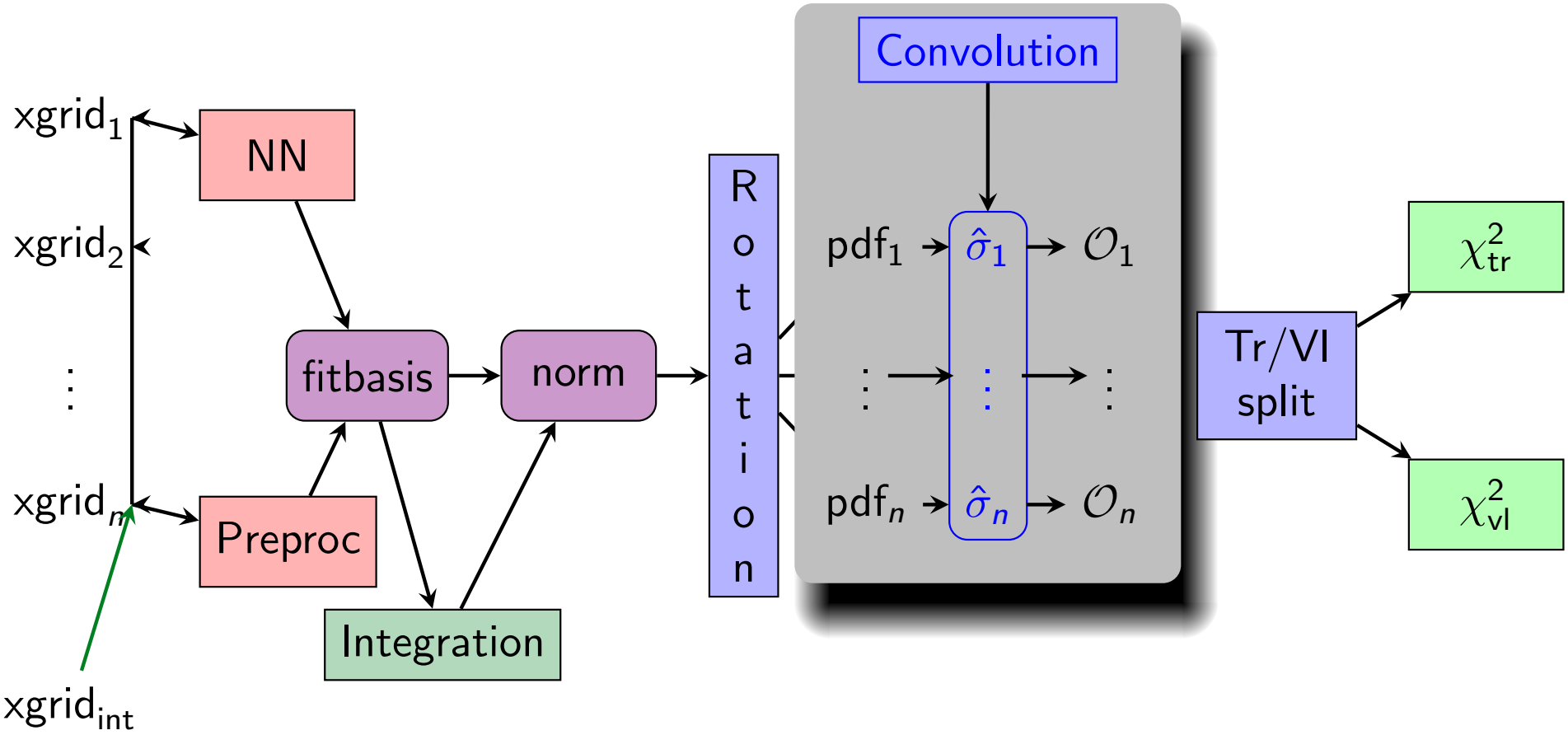
Tensorflow is very clever, but we have more information:



Customizing the operators

Tensorflow is very clever, but we have more information:

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



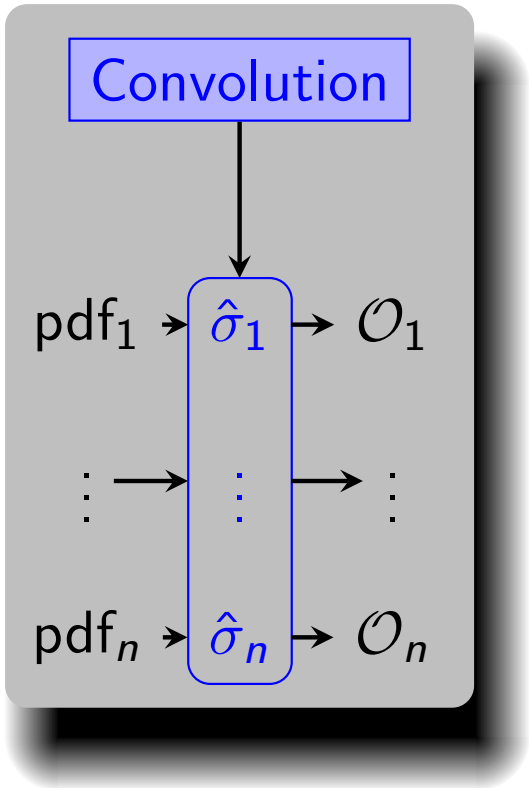
Customizing the operators

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
Time/epoch (s)	0.5 s	0.3 s

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



Hardware accelerating the fits

The problem of fitting many replicas is the perfect candidate for GPU parallelization

$\longrightarrow$ Not massively CPU intensive

$\longrightarrow$ Same operations are repeated for all replicas

Example operation, contraction of rank-2 tensors: $z_M^N = x_\alpha^N y_M^\alpha$.

N	M	α	CPU AVX	TF (CPU)	TF (GPU)	OpenCL (GPU)
8	10^3	10^5	0.48	0.44	0.552	1.10
8	10^4	10^5	4.86	4.13	4.68	3.41
$8 \cdot 10^3$	10^4	10^4	48.8	1.89	1.24	15.8

Comparison on the time-cost (in seconds) per operation

CPU in table corresponds to intel i9-9980XE

GPU in table corresponds to nvidia Titan V

Summary

✓ We have achieved a very powerful, flexible and fast machinery for PDF fitting.

✓ Faster run times: iterate over different choices of models or parameters.

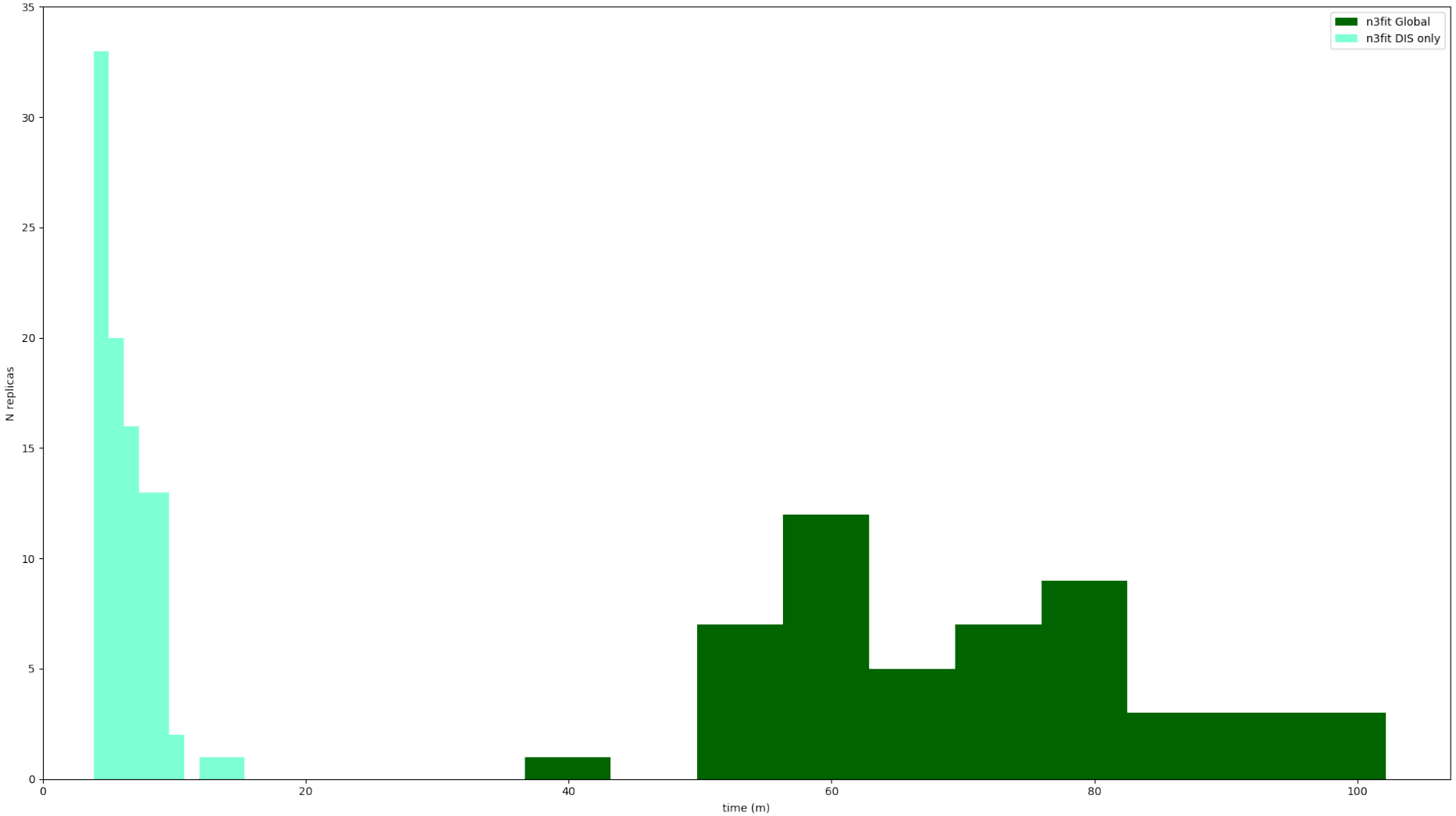
✓ The framework allows full customization *by design*.

⟶ **The cost of doing new studies is reduced, both the development/implementation and the raw computational cost.**

Future: can we also fit using FPGAs?

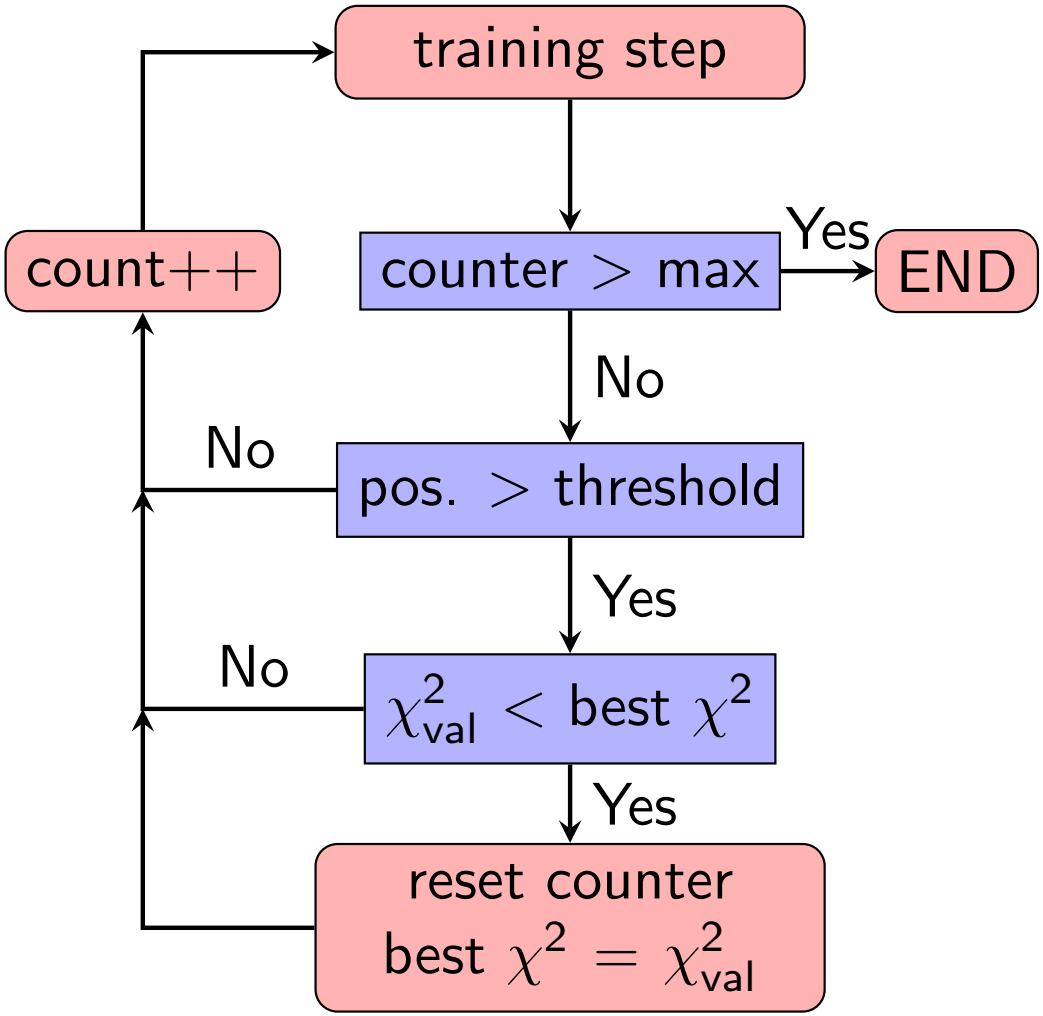
Thanks!

Zoom-in on the timings



Stopping

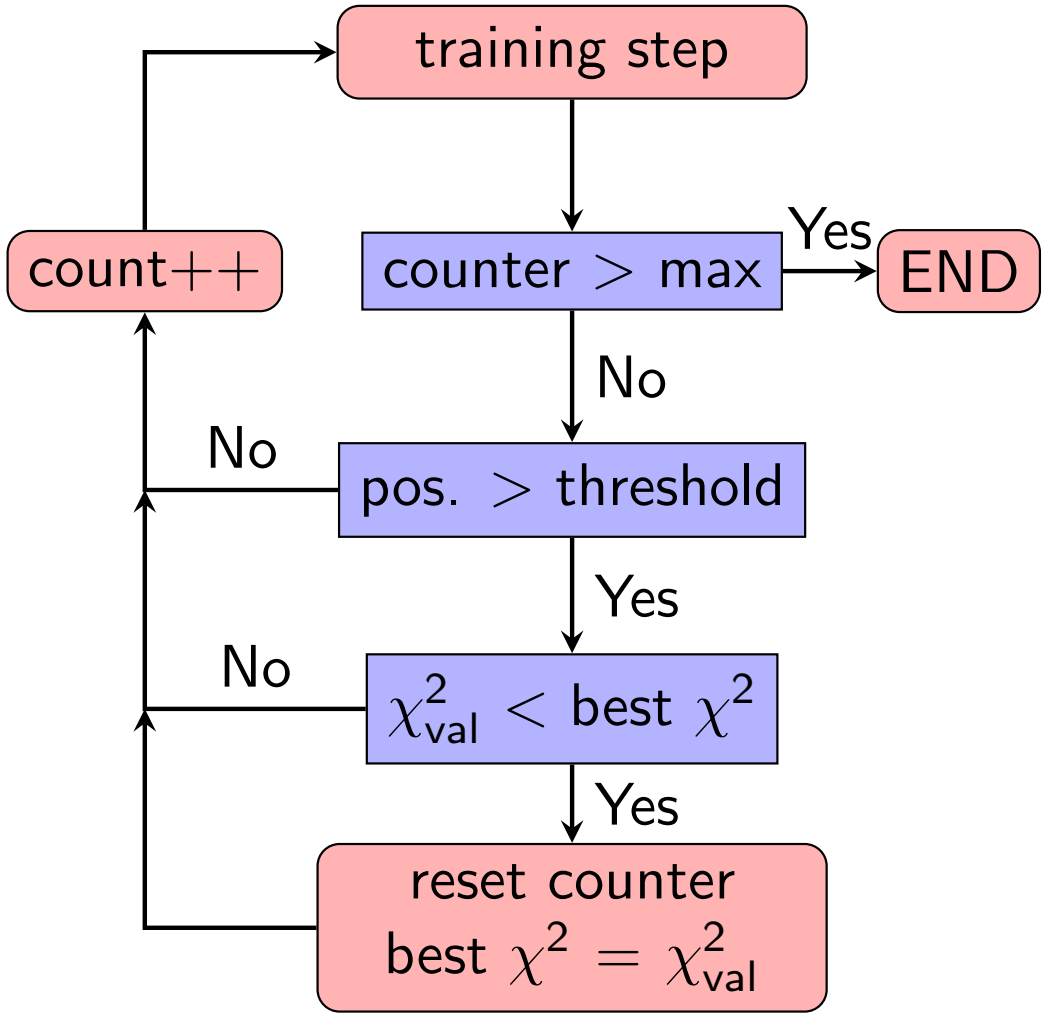
`Stopping method:` **Look-back method where positivity passes**



Stopping

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

Early stopping: reduce overfitting

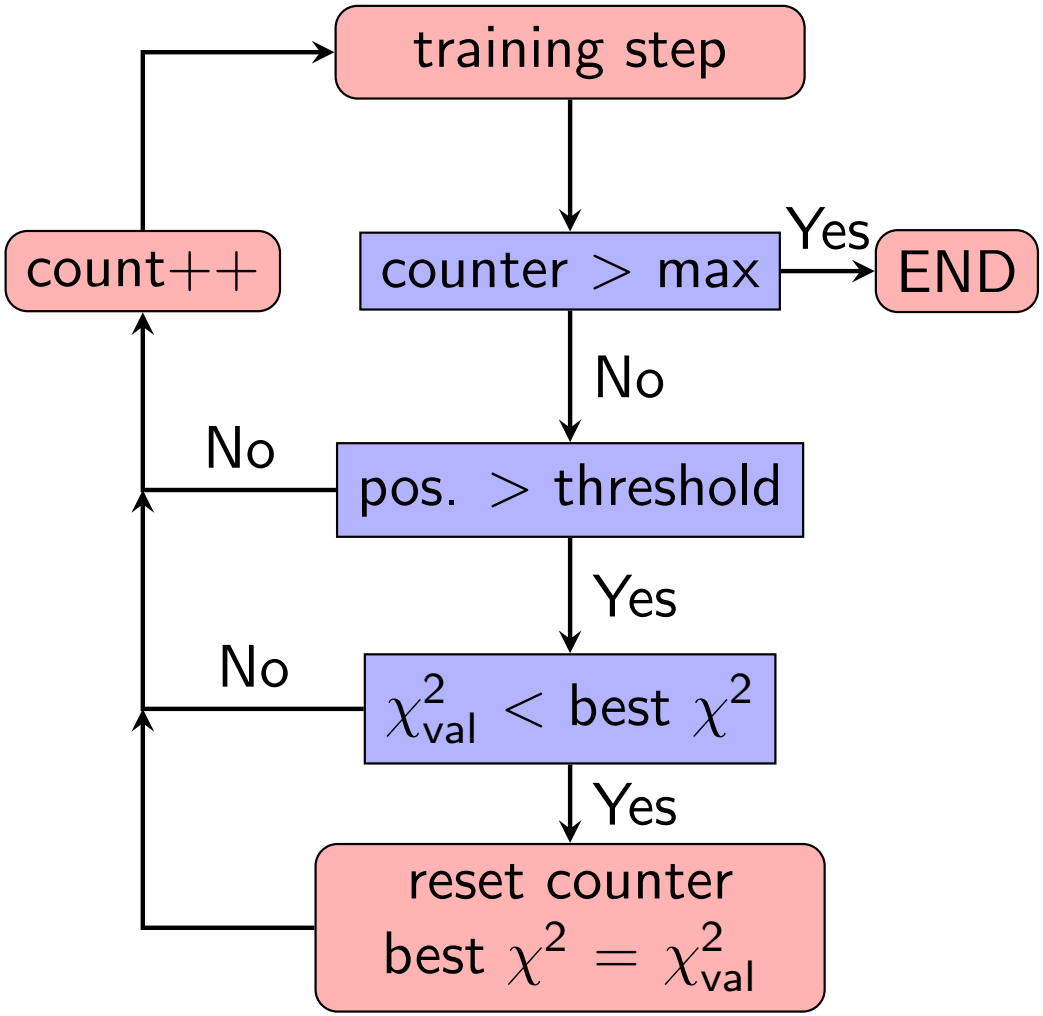


Stopping

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

Early stopping: reduce overfitting

- ✓ Minimize χ^2 of validation sets

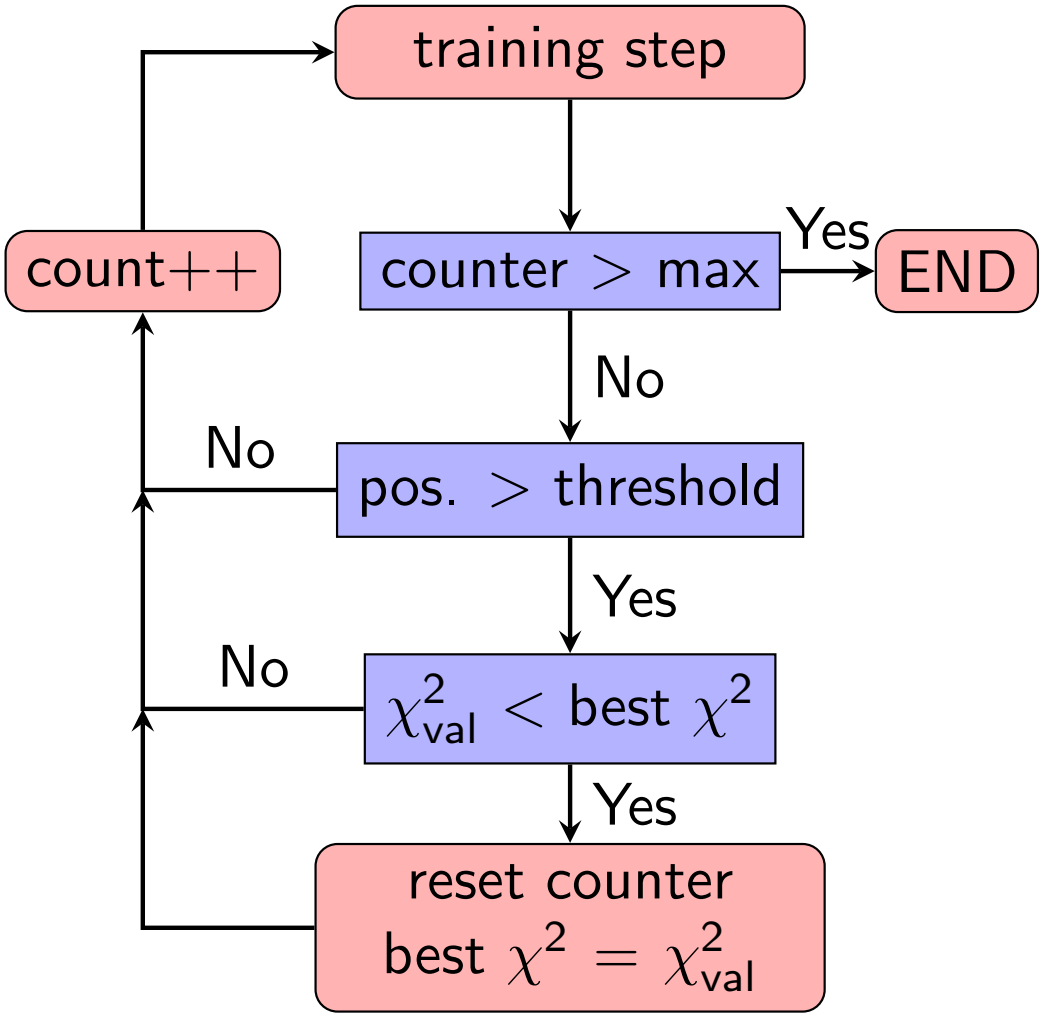


Stopping

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

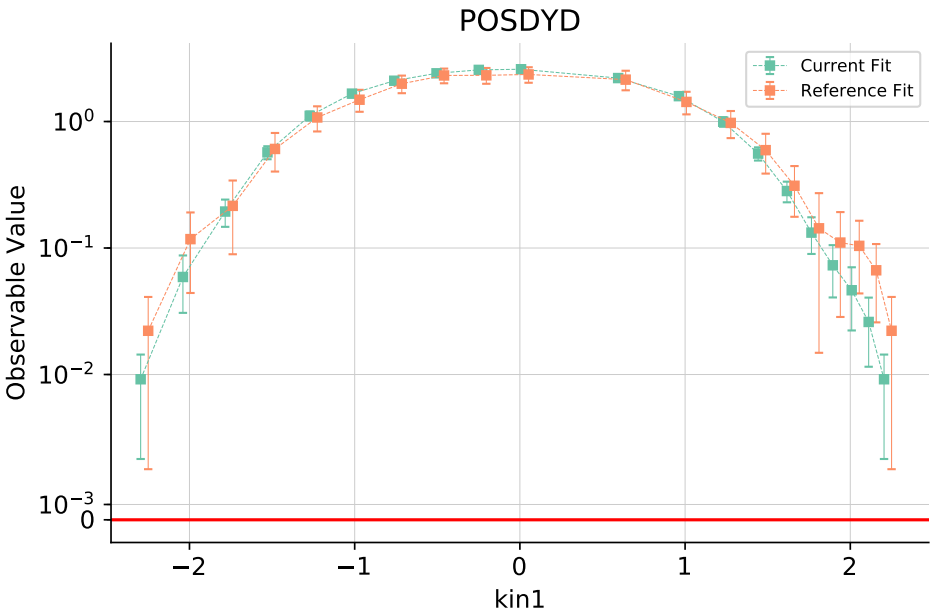
Early stopping: reduce overfitting

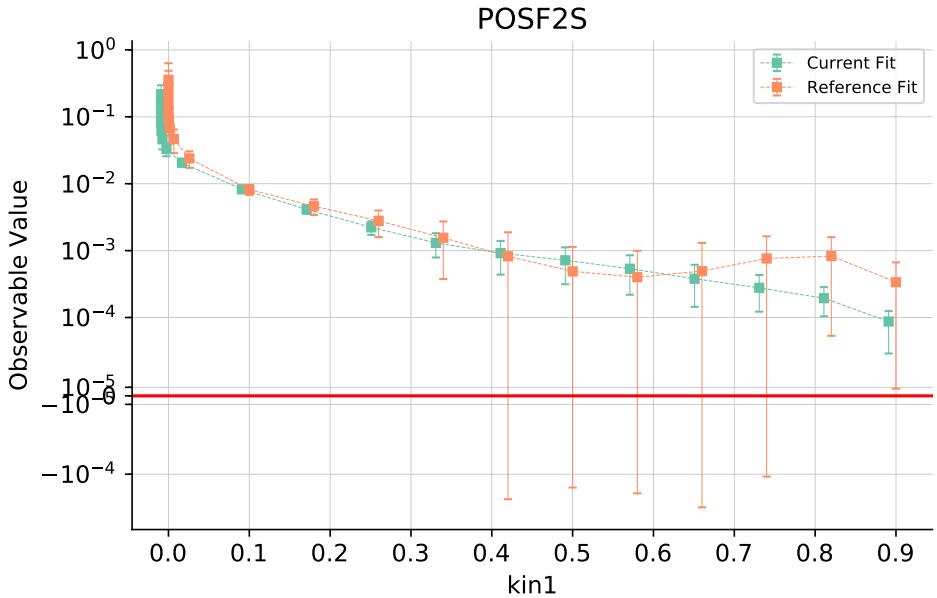
- ✓ Minimize χ^2 of validation sets
- ✓ Enforces positivity constraints



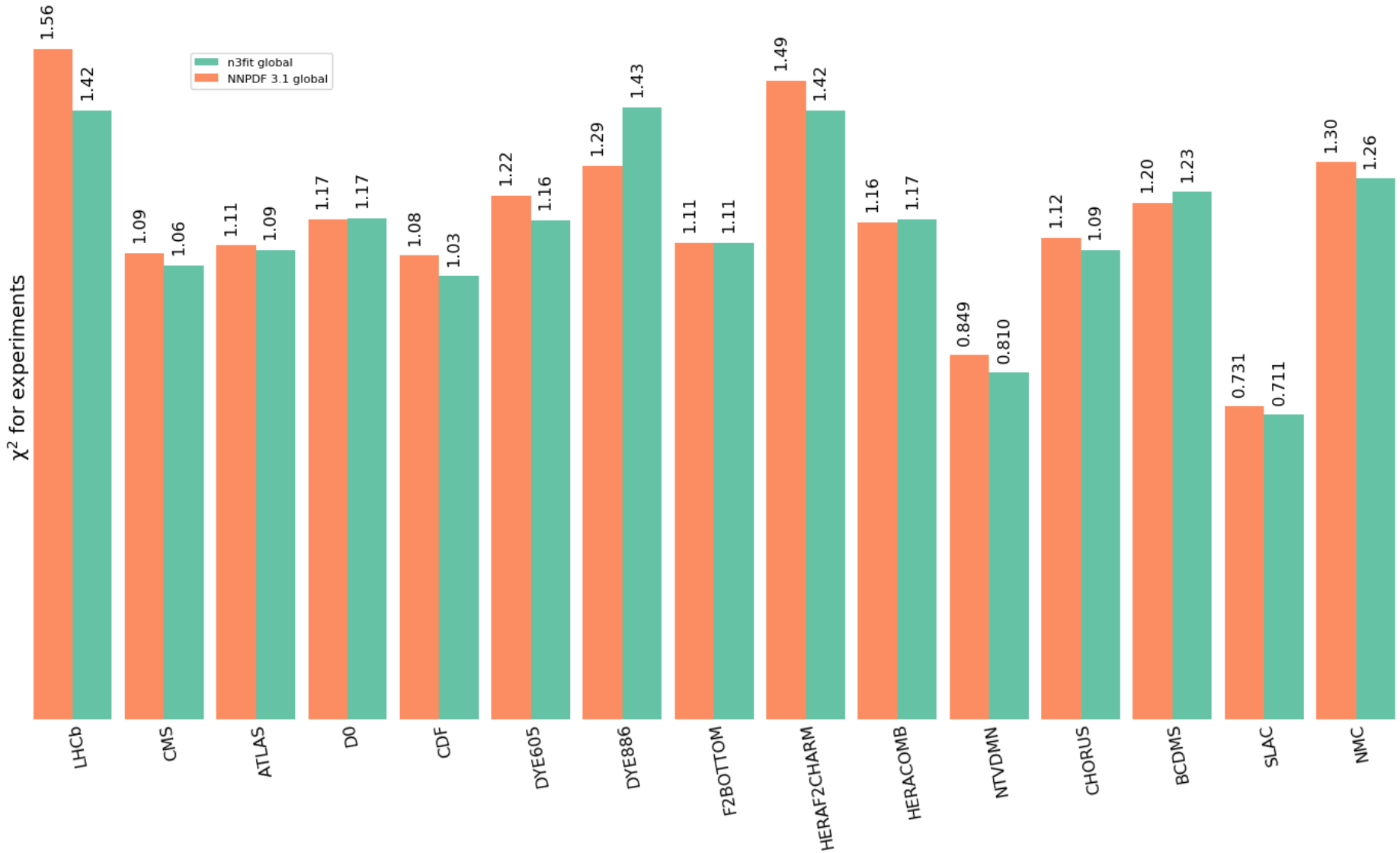
Positivity constrained

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

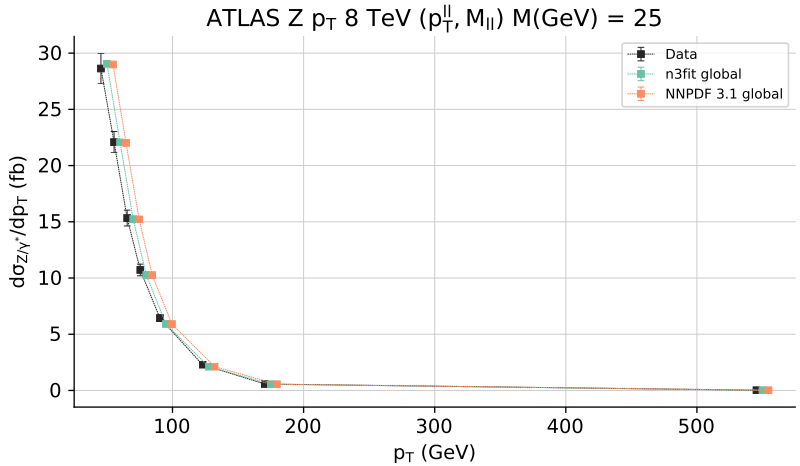


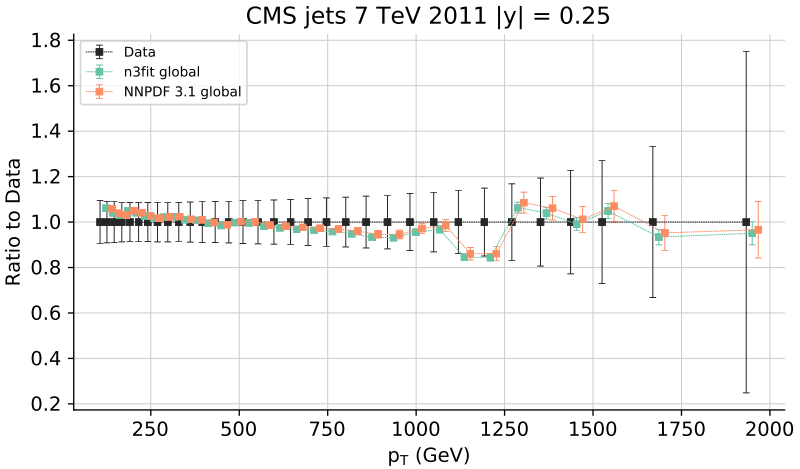


Per-experiment results



Comparison to data





→ Results compatible with NNPDF 3.1

→ Not only a similar χ^2-goodness but also similar per-point results

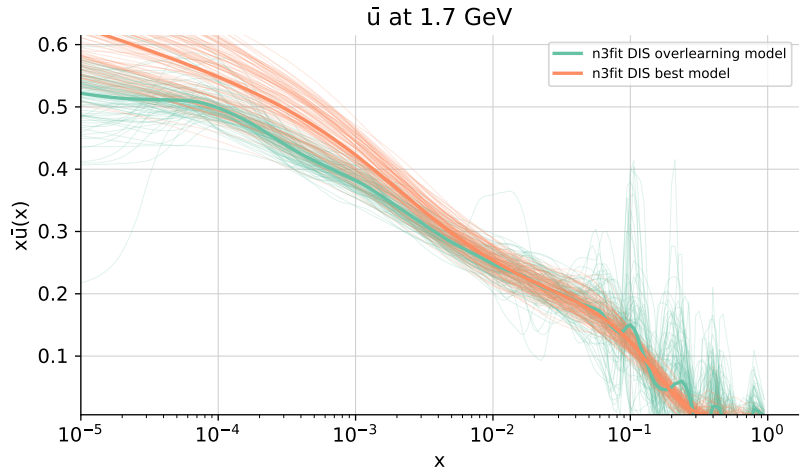
✓ The new methodology is compatible with the previous one!

Warning: overfitting!

With great power comes great responsability.

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.

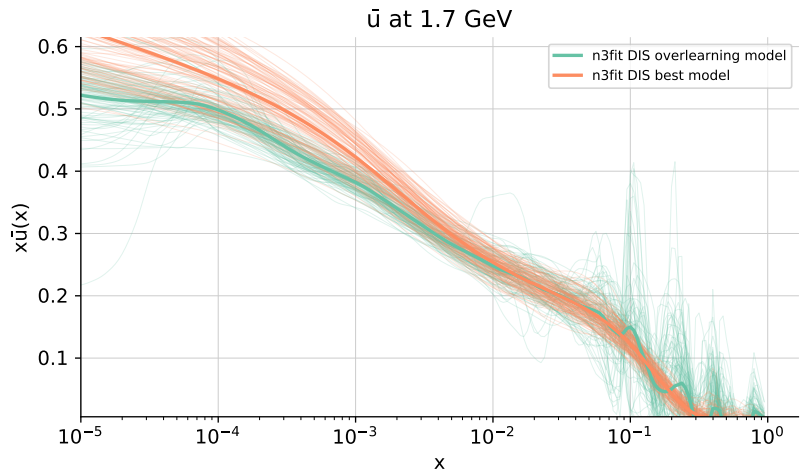


Warning: overfitting!

With great power comes great responsability.

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