Parton densities with deep learning models

based on arXiv:1907.05075

Stefano Carrazza

LHCP2020, 25 May 2020.

Università degli Studi di Milano and INFN Sezione di Milano

Acknowledgement: This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement number 740006.

PDF challenges

The PDF priorities

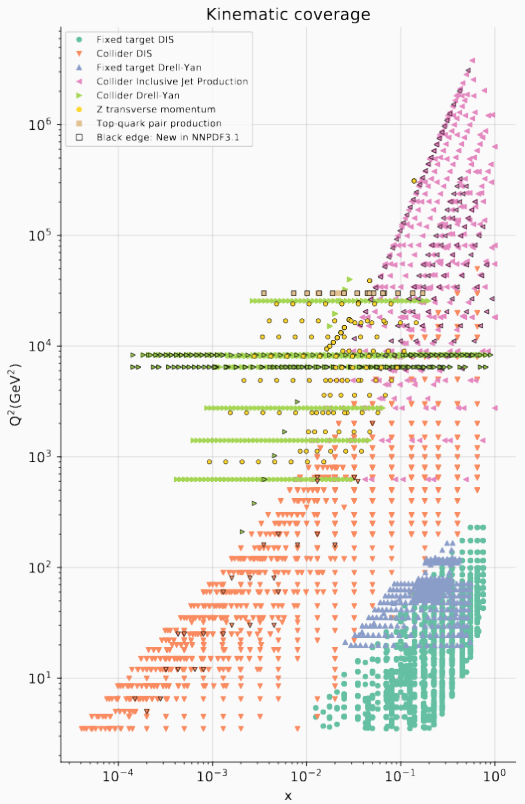
Parton distribution functions have been developed in a systematic way from the 2000s. Since that time, the PDF community gave priority to:

The PDF priorities

Parton distribution functions have been developed in a systematic way from the 2000s. Since that time, the PDF community gave priority to:

Data

Collect and implement data from different processes.



The PDF priorities

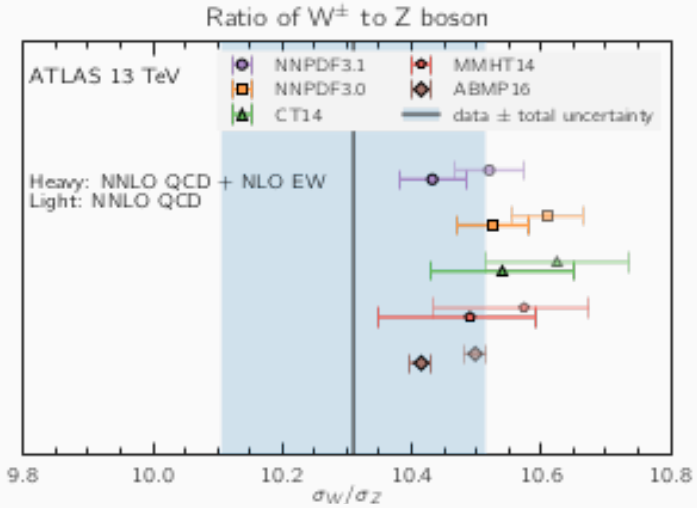
Parton distribution functions have been developed in a systematic way from the 2000s. Since that time, the PDF community gave priority to:

Data

Collect and implement data from different processes.

Theory

Compute theoretical predictions for multiple processes.



The PDF priorities

Parton distribution functions have been developed in a systematic way from the 2000s. Since that time, the PDF community gave priority to:

Data

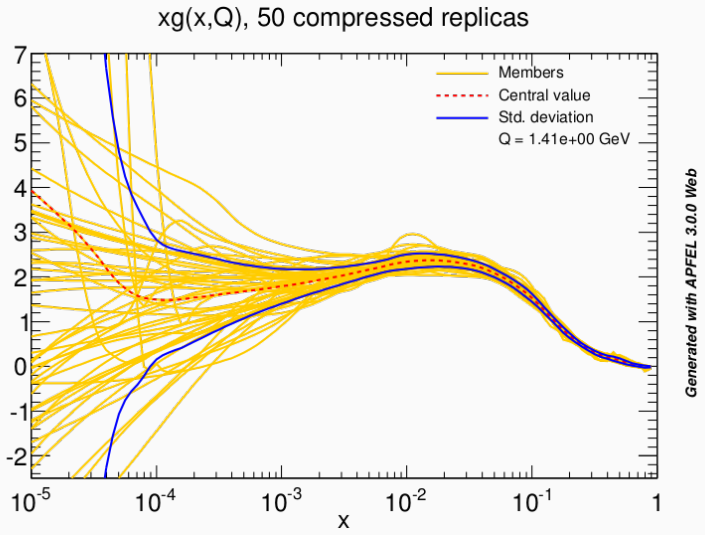
Collect and implement data from different processes.

Theory

Compute theoretical predictions for multiple processes.

Methodology

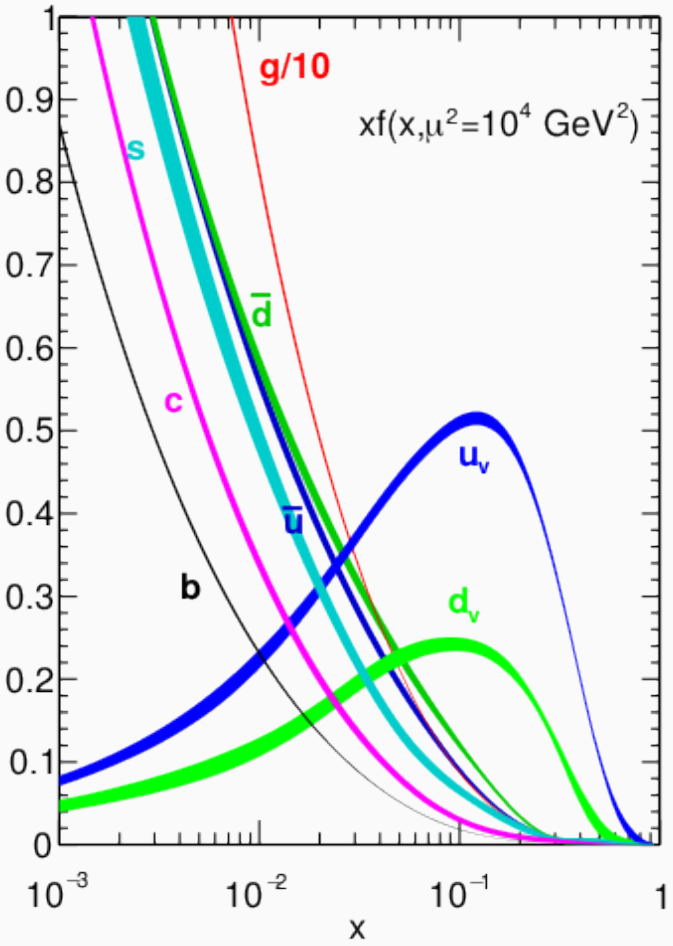
Define an optimized regression models for the PDF fits.



The NNPDF methodology

The technology used in **NNPDF3.1**:

- Neural Networks optimized with Genetic Algorithms
- Custom implementation in C++
- Tuning performed manually



PDF challenges

Challenges:

- How to increase **fit performance speed**?
  - faster fits $\Rightarrow$ more fits
- How can we **tune/learn the methodology**?
  - select the best model for our data/theory

PDF challenges

Challenges:

- How to increase **fit performance speed**?
  - faster fits $\Rightarrow$ more fits
- How can we **tune/learn the methodology**?
  - select the best model for our data/theory

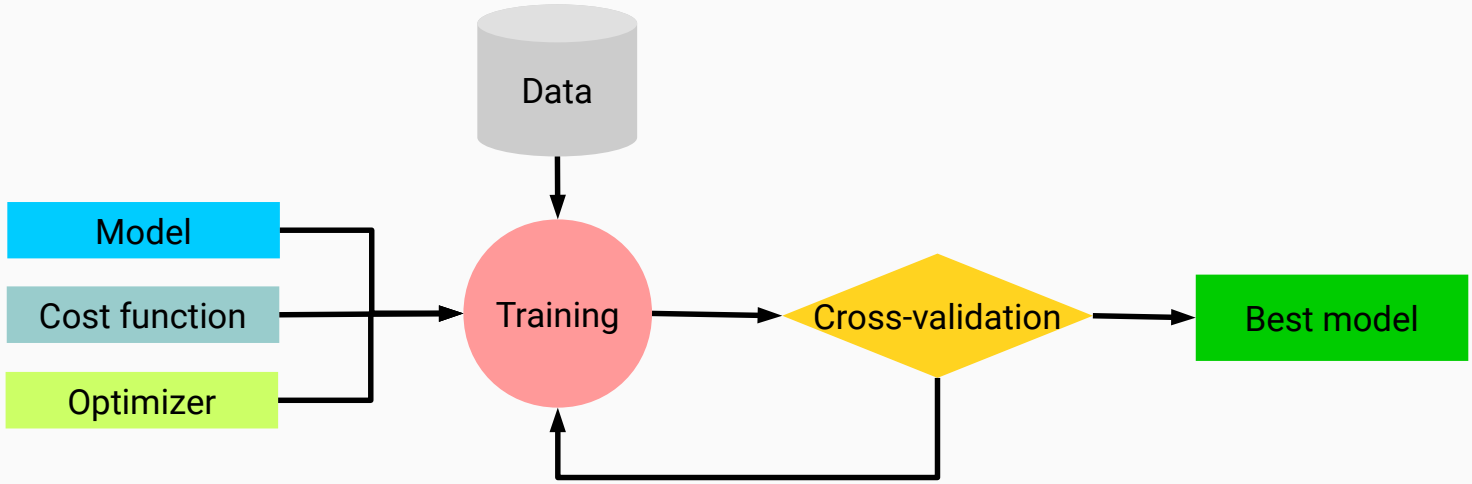
Solution $\Rightarrow$ move towards **deep learning**

- in terms **software/technology**
- in terms of **methodology**

Towards a DL approach

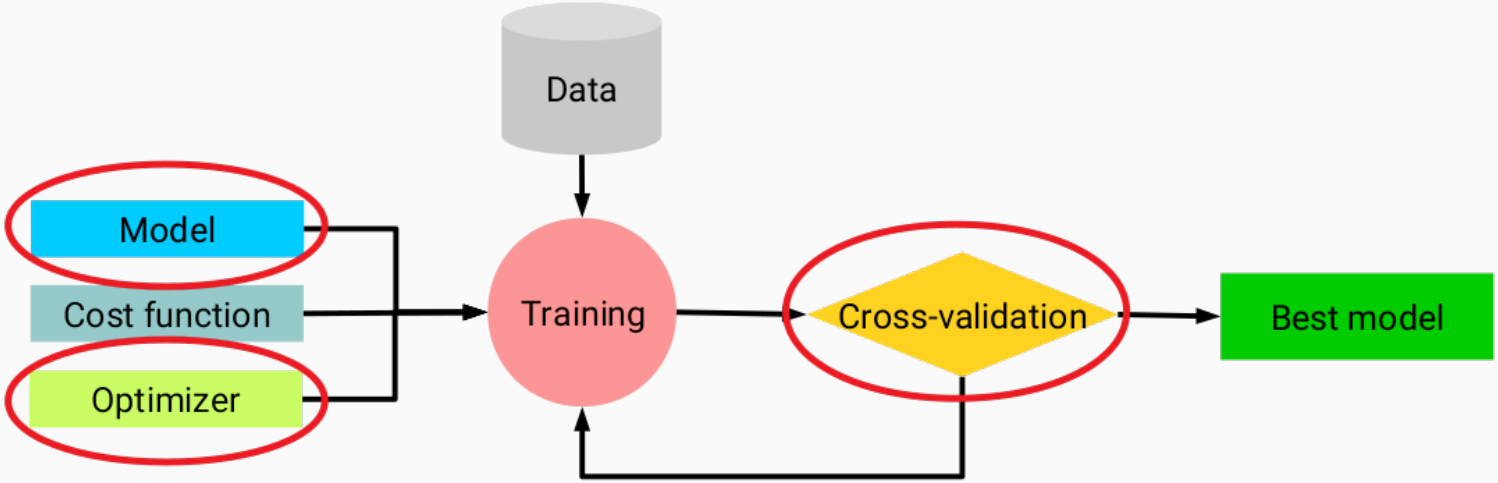
Deep learning pipeline

PDF determination is a supervised learning problem thus we need to provide review for the following sectors:

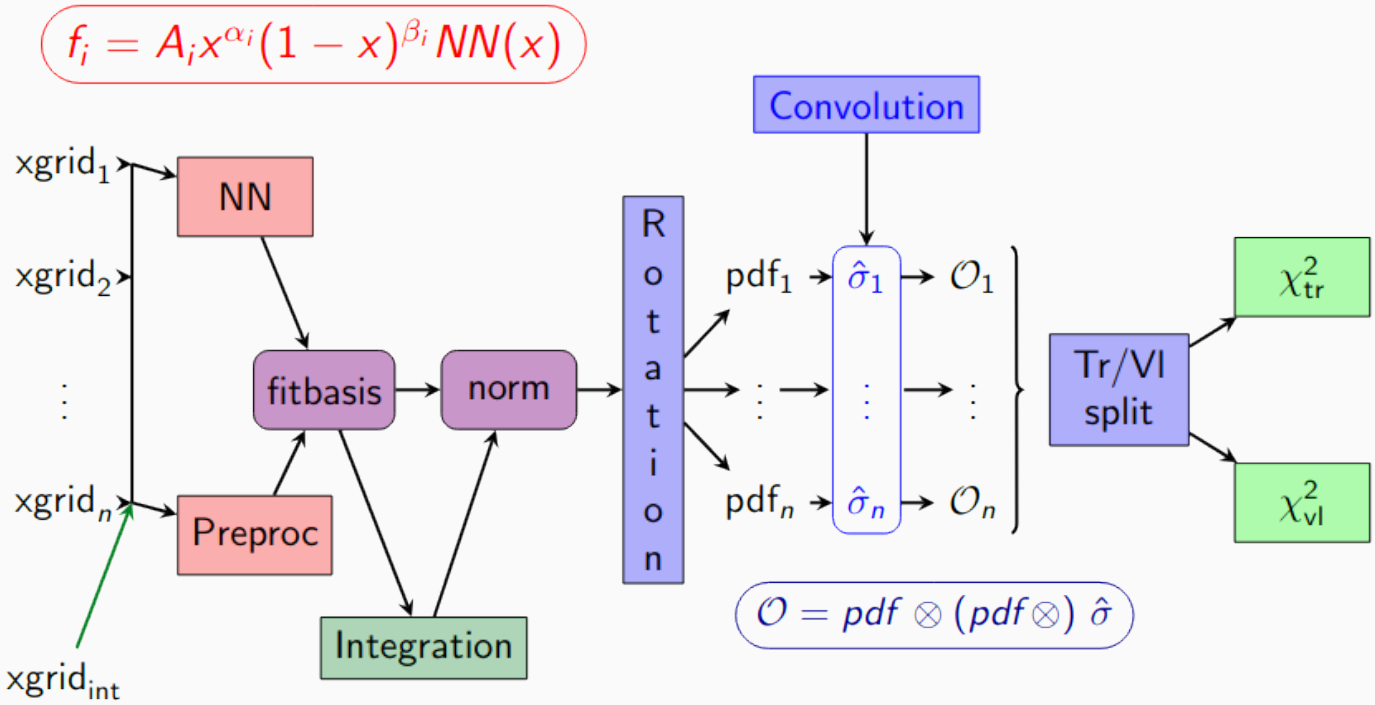


Deep learning pipeline

PDF determination is a supervised learning problem thus we need to provide review for the following sectors:



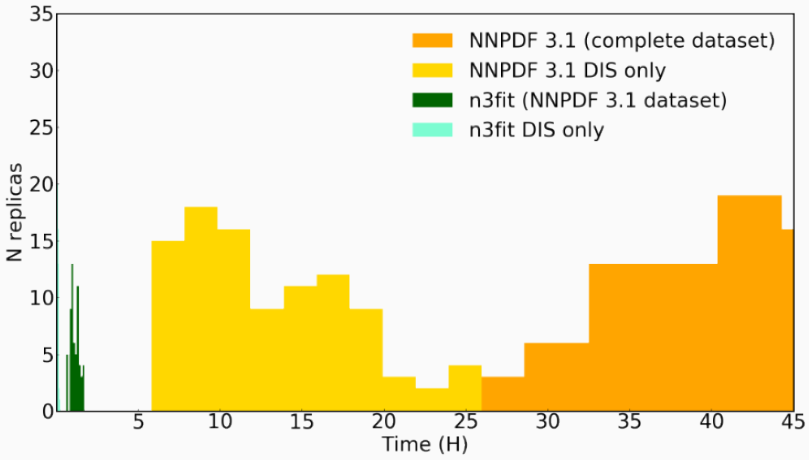
The `n3fit` model

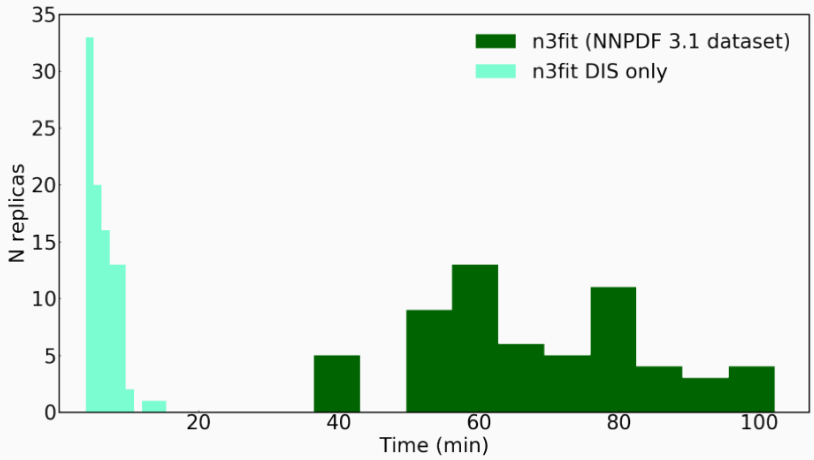


New features:

- `Python/C++` implementation using **TensorFlow**
- Modular approach $\Rightarrow$ easier and faster development
- Can vary all aspects of the methodology

Performance benefits - time per replica





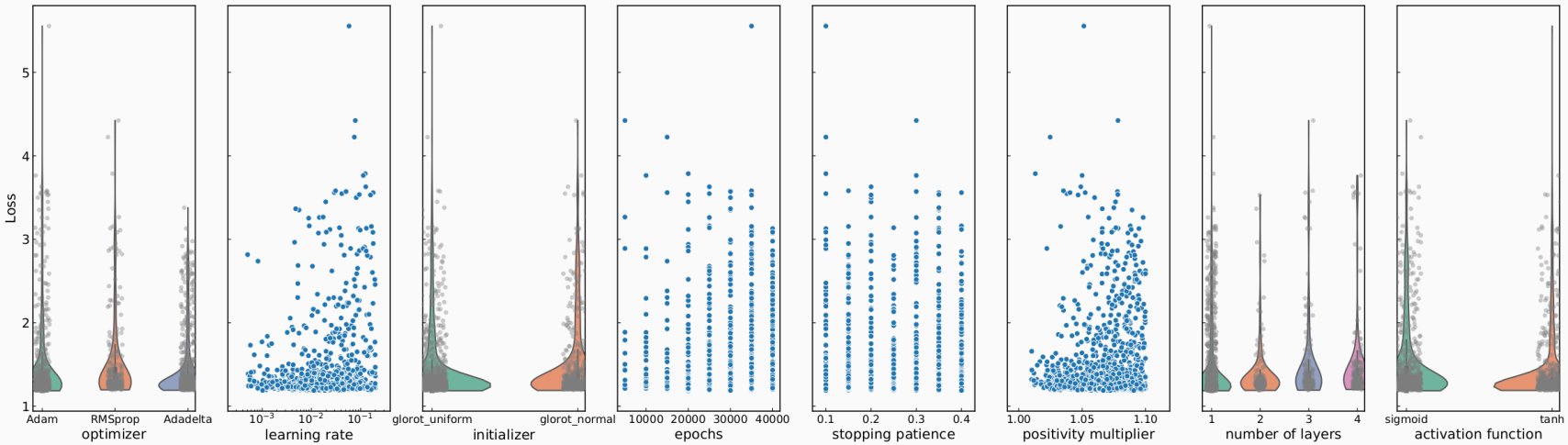
Benefits

- Gain on speed and efficiency, less **CPU hours for a fit**
- Usage of new technologies $\rightarrow$ hardware, libraries
- Usage of **gradient descent** optimization methods

$\Rightarrow$ **Possibility to learn and tune the methodology**

Learning the methodology

How to determine the best methodology?



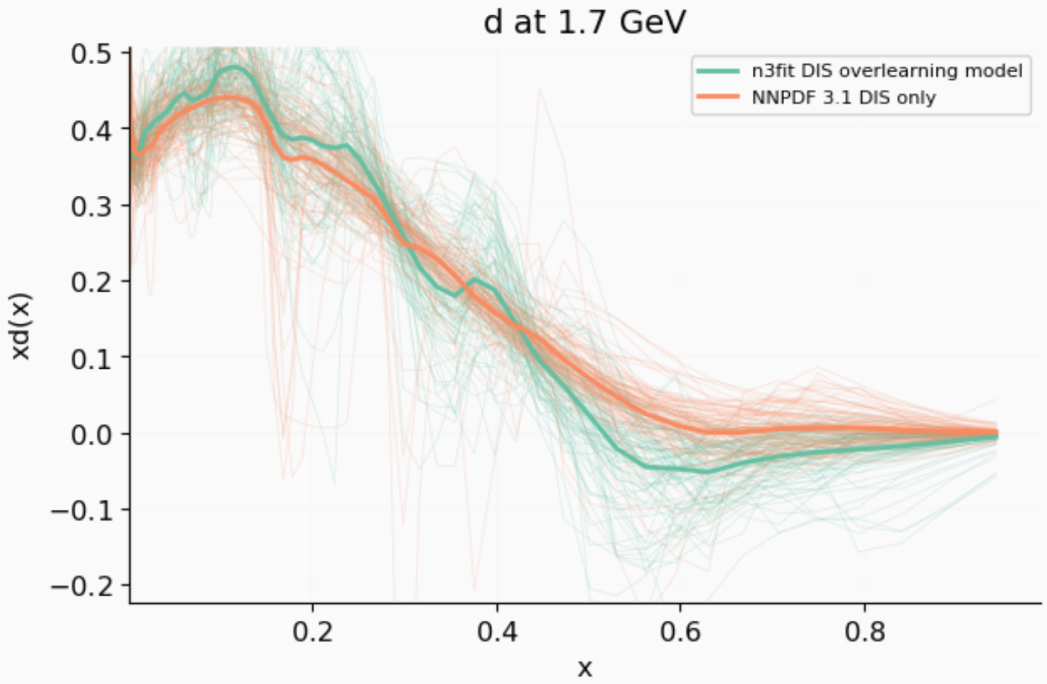
Perform **hyperoptimization scans**:

Neural Network	Fit options
Number of layers (*)	Optimizer (*)
Size of each layer	Initial learning rate (*)
Dropout	Maximum number of epochs (*)
Activation functions (*)	Stopping Patience (*)
Initialization functions (*)	Positivity multiplier (*)

- Optimize figure of merit: validation χ^2
- Use bayesian updating (`hyperopt`)

The overfitting problem

Using validation set χ^2:

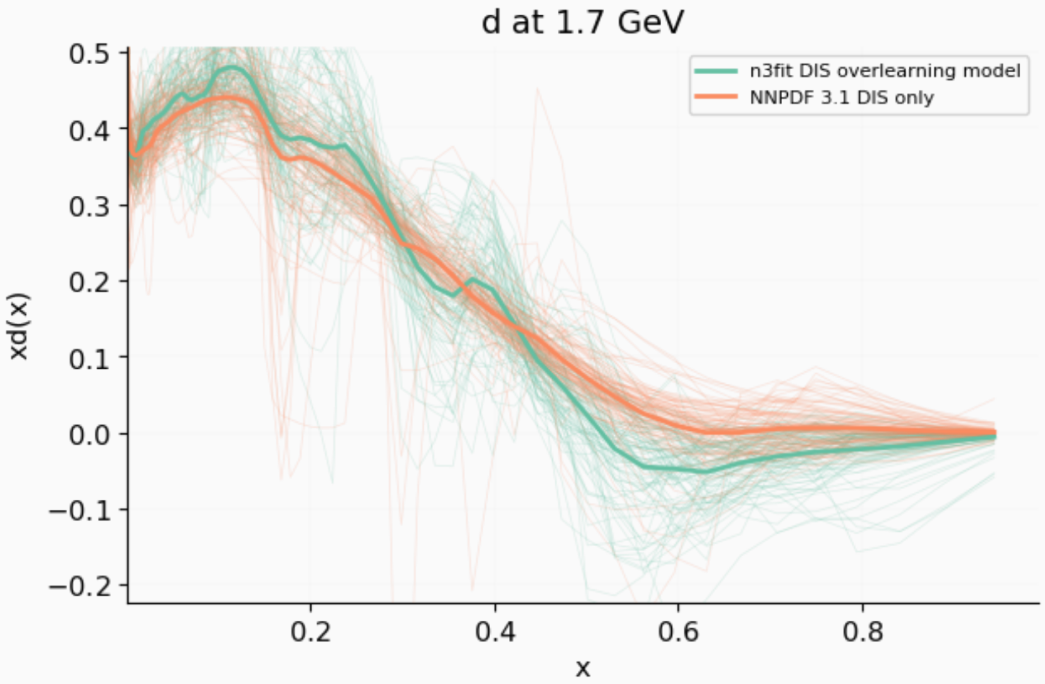


The choice of the **right figure of merit** is important:

- **NNPDF** wiggles $\rightarrow$ finite size , goes away as $N_{\rm rep}$ grows
- **N3PDF** wiggles $\rightarrow$ **overfitting**, correlations training-validation data!

The overfitting problem

Using validation set χ^2:



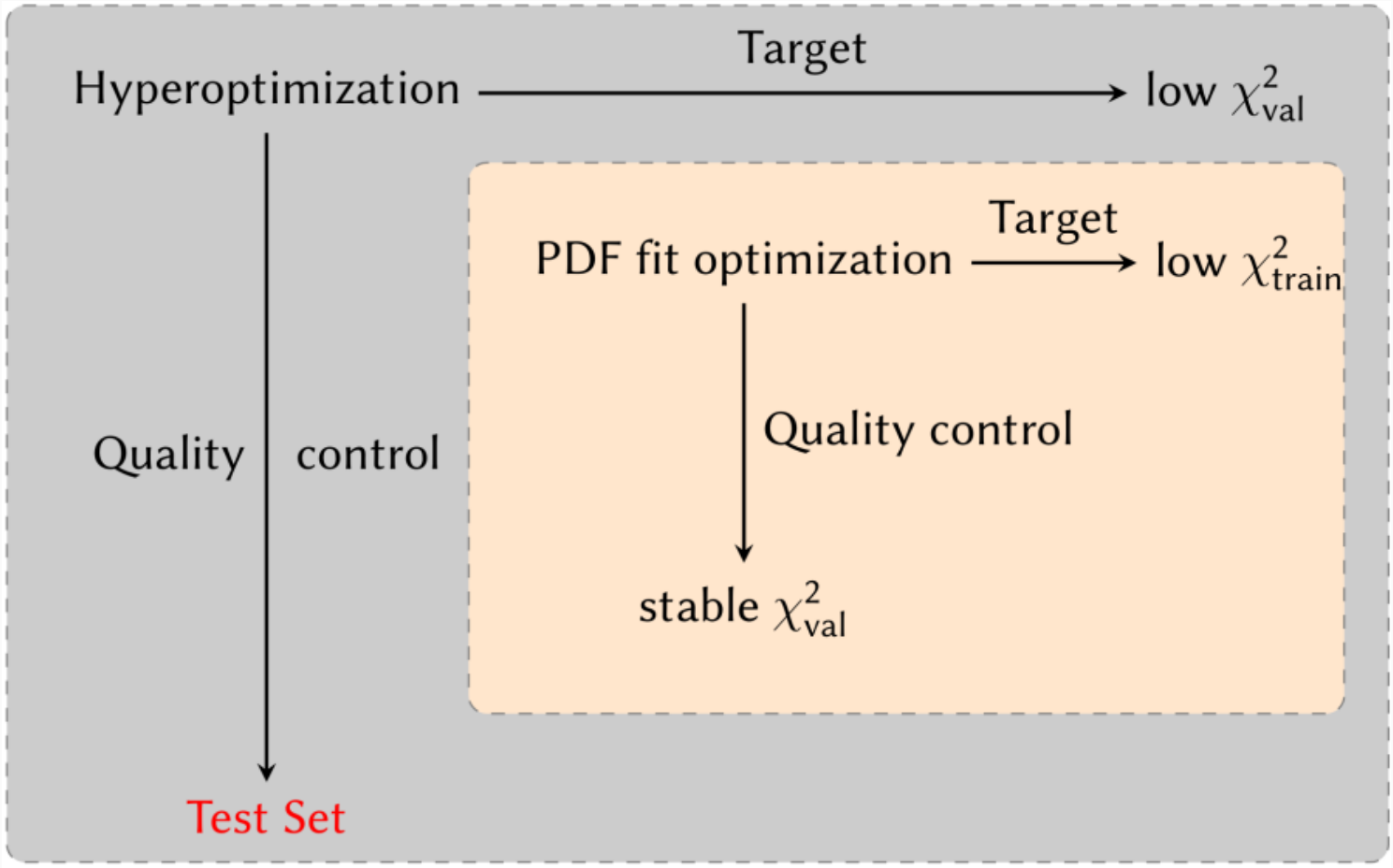
The choice of the **right figure of merit** is important:

- **NNPDF** wiggles $\rightarrow$ finite size , goes away as $N_{\rm rep}$ grows
- **N3PDF** wiggles $\rightarrow$ **overfitting**, correlations training-validation data!

$\Rightarrow$ **define a proper quality control criterion**

Cross-Validation vs hyperoptimization

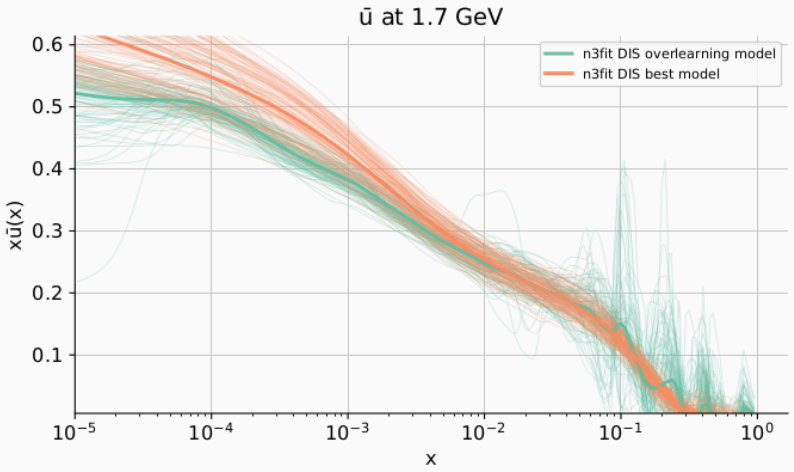
Define a completely uncorrelated **Test Set**

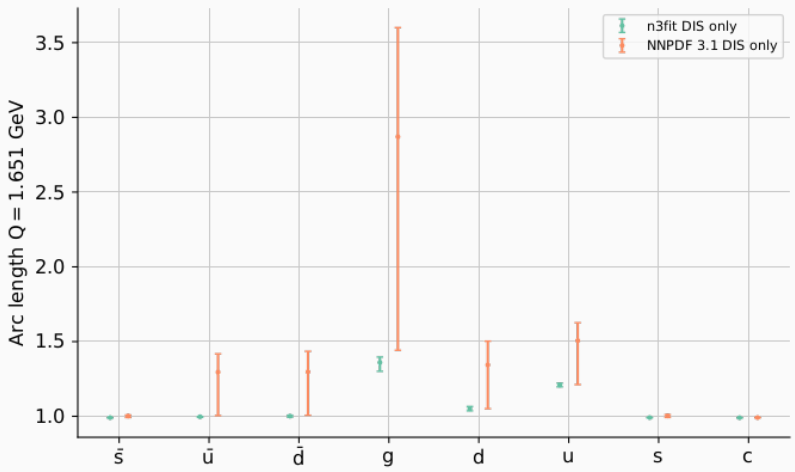


Optimize on weighted average of validation and test.

Removing overfitting

Using test-validation set χ^2:





- No **overfitting**
- Greater **stability**
- Reduced uncertainties

	DIS only	Global
`n3fit` (new)	1.10	1.15
`nnfit` (old)	1.13	1.16

Quality control

Chronological fits

Idea:

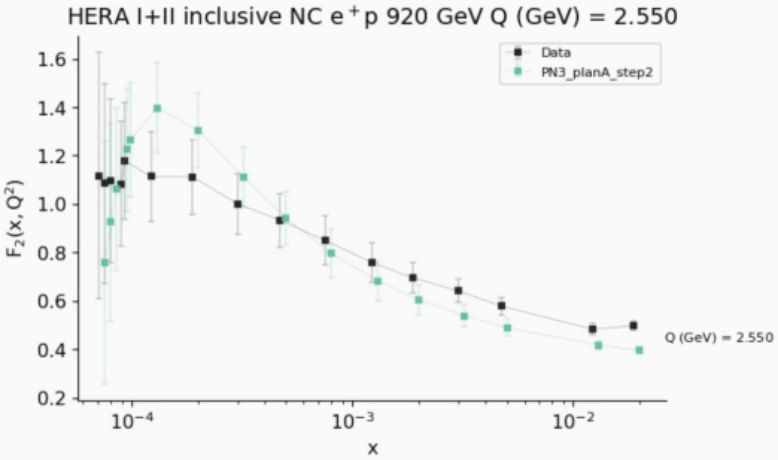
1. Take a pre-HERA dataset
2. Perform hyperoptimization
3. Compare predictions to "future" data

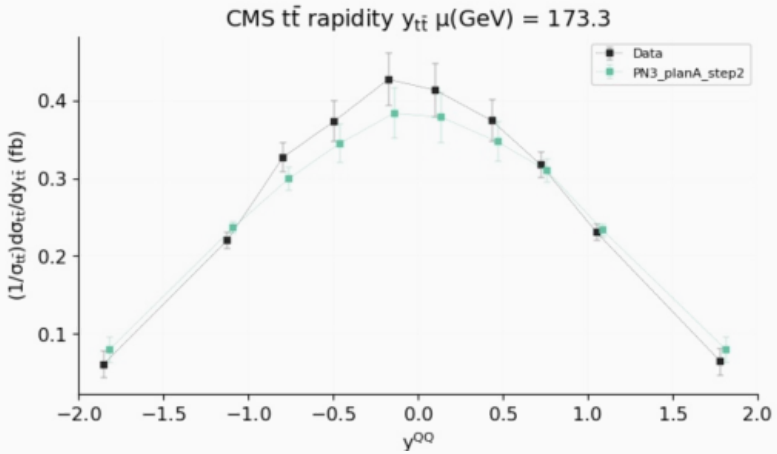
Chronological fits

Idea:

1. Take a pre-HERA dataset
2. Perform hyperoptimization
3. Compare predictions to "future" data

Examples:





⇒ **Results within PDF uncertainty!**

Defining a proper Test set

How to define a proper Test Set?

- we have a limited dataset with lots of features, $N_{\text{data}} \approx 5000$

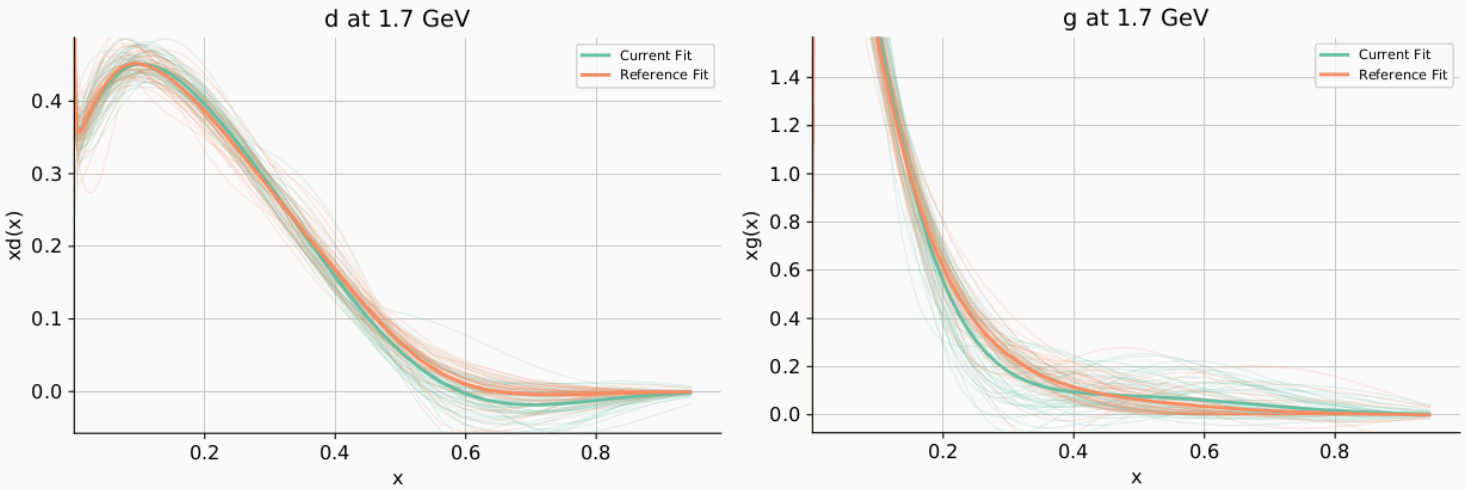
Defining a proper Test set

How to define a proper Test Set?

- we have a limited dataset with lots of features, $N_{\text{data}} \approx 5000$

$\Rightarrow$ **Potential solution:** use k-fold cross-validation.

- Use k partitions in a rotation estimation for the Test Set
- hyperoptimize the mean value of the Test Set χ^2



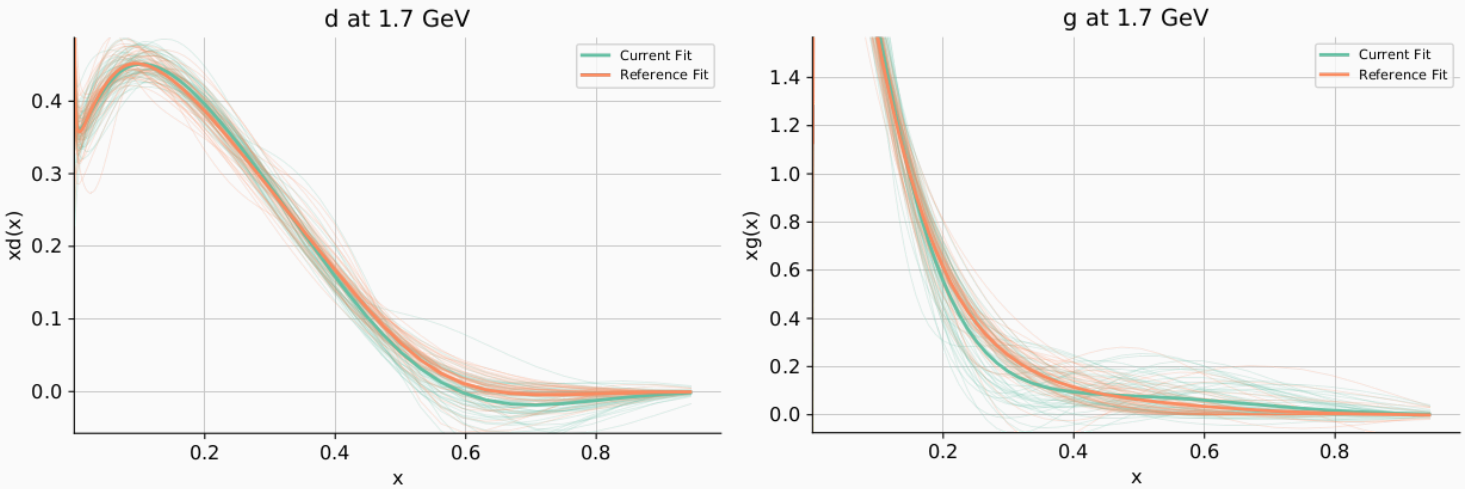
Defining a proper Test set

How to define a proper Test Set?

- we have a limited dataset with lots of features, $N_{\text{data}} \approx 5000$

$\Rightarrow$ **Potential solution:** use k-fold cross-validation.

- Use k partitions in a rotation estimation for the Test Set
- hyperoptimize the mean value of the Test Set χ^2



$\Rightarrow$ **Compatible with our previous Test Set definition.**

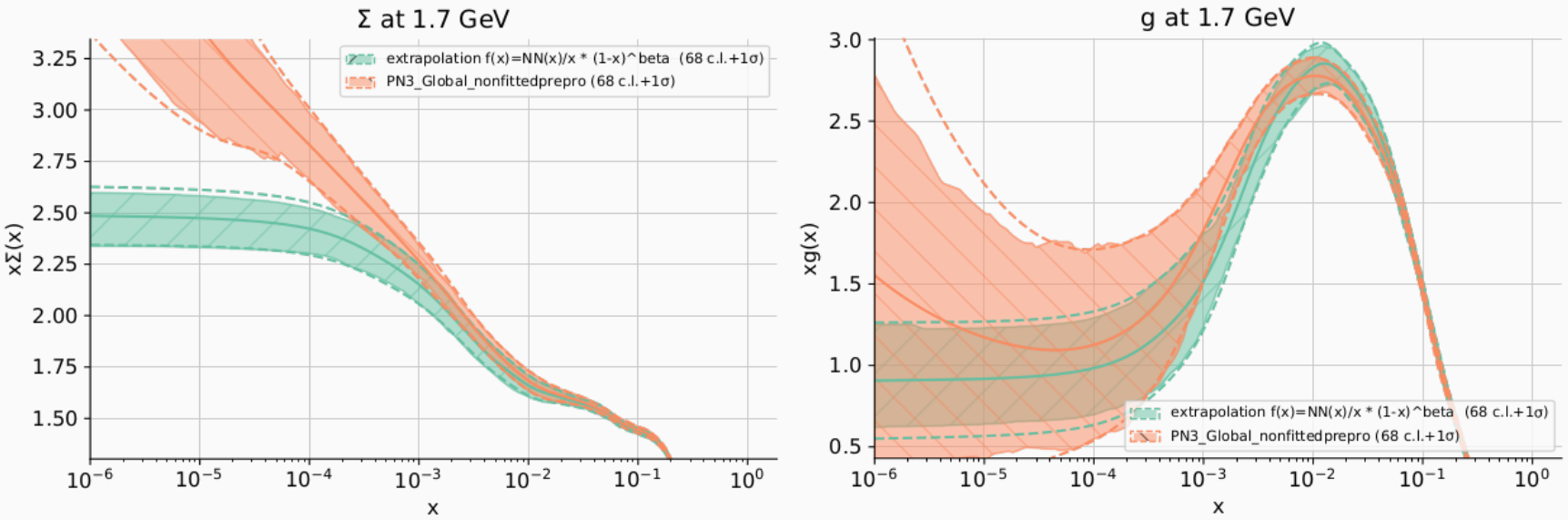
Future challenges

Extrapolation region

The current parametrization uses preprocessing:

$$f(x) = x^{-\alpha}(1-x)^{\beta} NN(x)$$

If preprocessing is removed, we observe saturation at small-x:



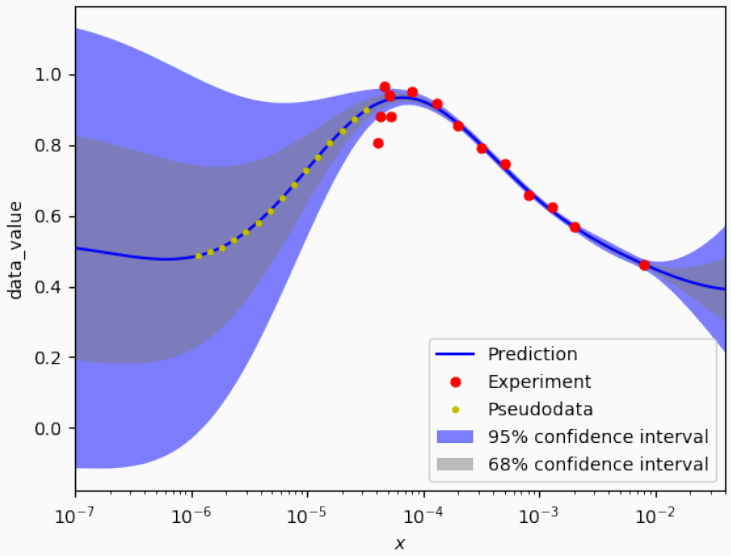
Challenges:

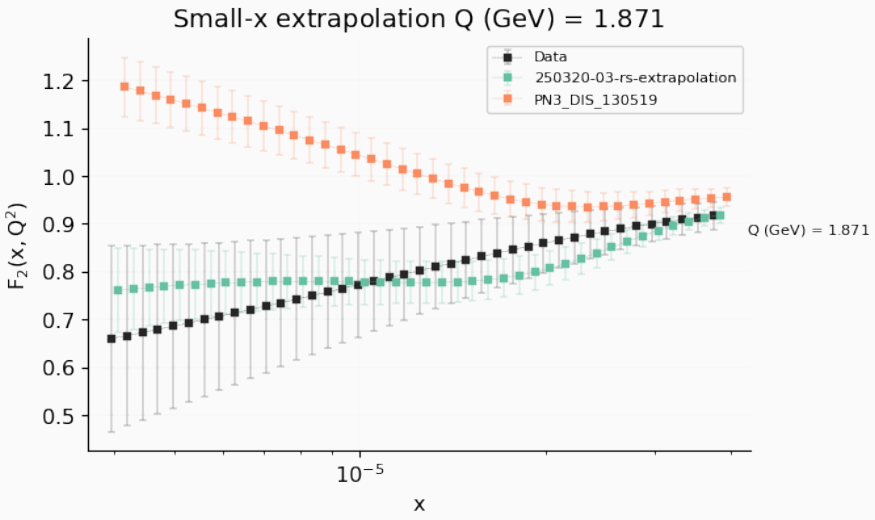
- Modify neural network input architecture
- Generate pseudodata in the extrapolation region

Extrapolation region

Gaussian pseudodata:

- use gaussian process to model DIS observables
- propagate a prior gaussian into extrapolation
- generate gaussian pseudodata and add it to fit





Final comments

Towards the NNPDF4.0 release:

- Faster run times and stable results
- Possibility to learn the methodology
- Quality control, reduced uncertainties
- Better understanding of model behavior

Thank you!