Fragmentation Functions and Global QCD Fits

Lecture 2: Methodological aspects of global QCD fits

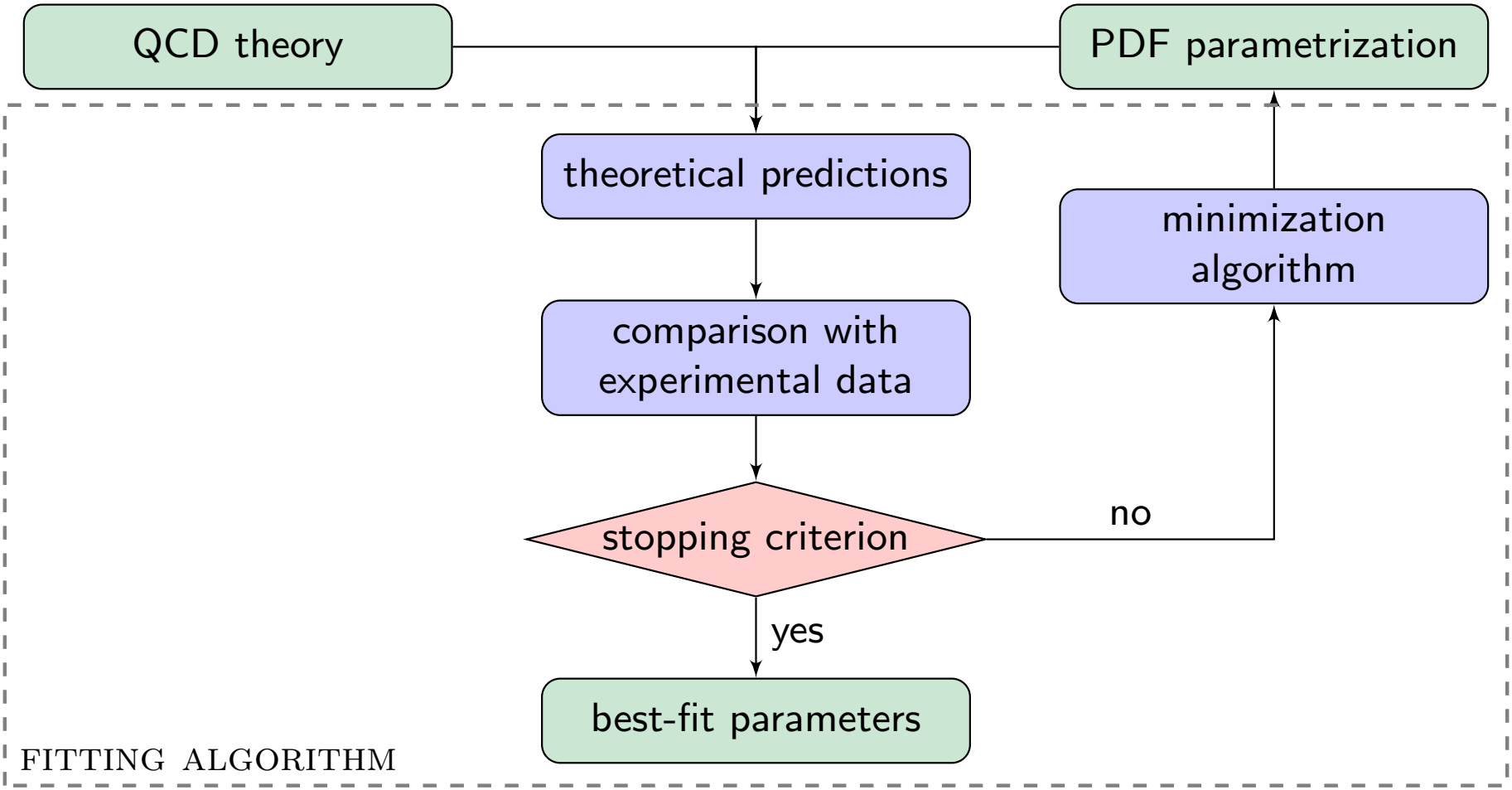
Outline

1. A global fit: the problem
   determining the probability density in a space of functions
2. Parton parametrisation
   standard and flexible
3. Goodness of fit
   dealing with various sources of experimental uncertainties
4. Parameter optimisation
   determining the best-fit configuration of parameters
5. Representation of uncertainties
   Hessian and Monte Carlo
6. Closure tests
   tracing the source of uncertainty
7. Reweighting
   assessing the impact of new data without refitting

Methods and techniques developed and widely used for fits of unpolarised PDFs
The discussion can be easily extended to FFs

2.1 A global fit: the problem

A global FF determination: the underlying strategy

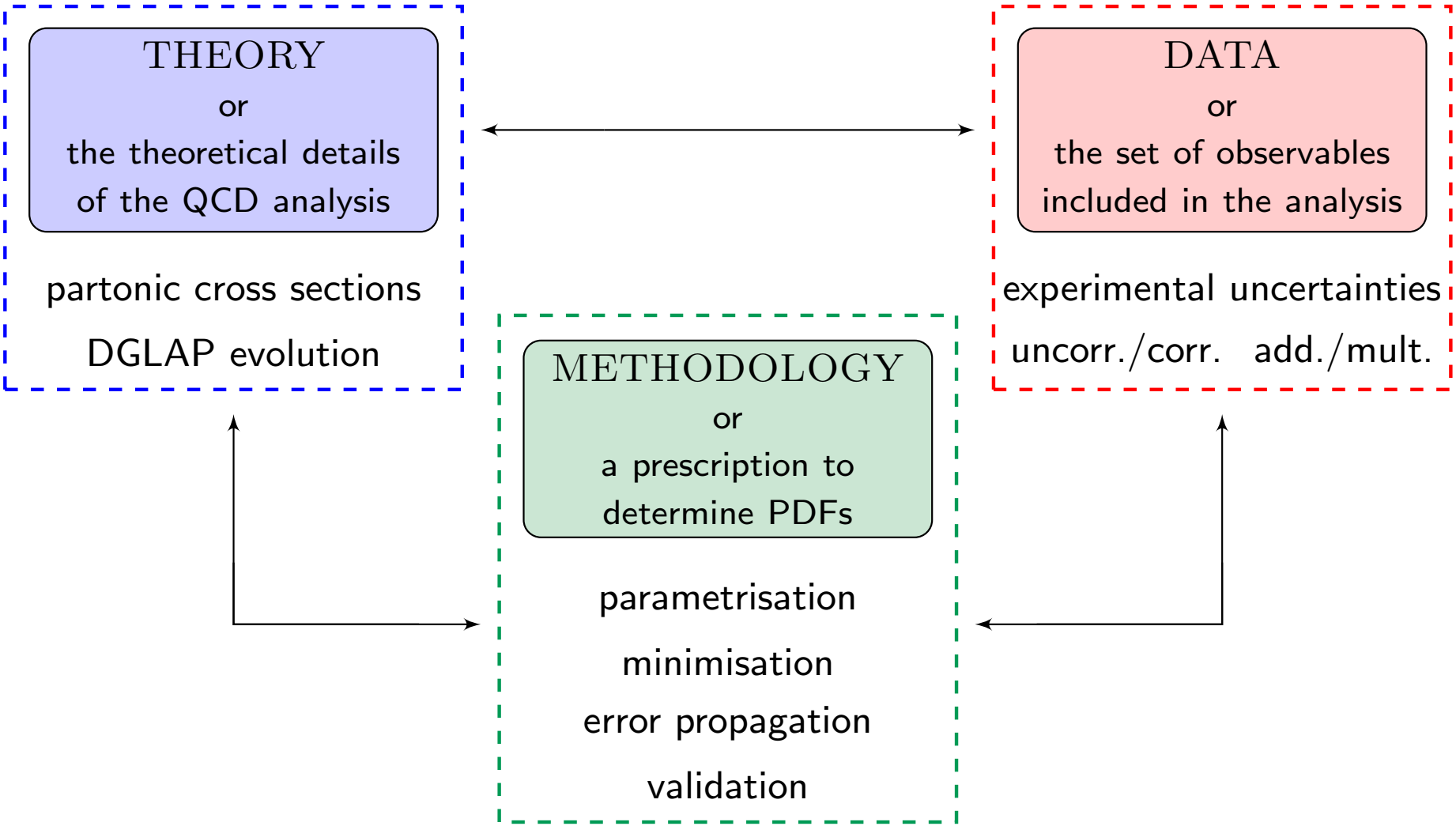


Assume a reasonable PDF parametrization

Obtain theoretical predictions for various processes and compare predictions to data

Determine the best-fit parameters via minimization of a proper figure of merit

A global FF determination: the ingredients we need



Each of these ingredients is a source of uncertainty in the FF determination

A global FF determination: an ill-posed problem

For a single quantity, we quote one-sigma errors: value $\pm$ error

For a pair of numbers, we quote a one-sigma ellipse

For a function, we need a multi-dimensional *error bar* in the space of functions

We must determine the probability density (measure) $\mathcal{P}[f]$
in the space of of fragmentation functions $[f]$

For any observable $\mathcal{O}$ depending on a set of fragmentation functions $[f]$
its expectation value and uncertainty are functional integrals over the space of FFs

$$\langle \mathcal{O}[f] \rangle = \int \mathcal{D}f \, \mathcal{P}[f] \, \mathcal{O}[f] \qquad \text{expectation value}$$

$$\sigma_{\mathcal{O}}[f] = \left[\int \mathcal{D}f \, \mathcal{P}[f] \, \left(\mathcal{O}[f] - \langle \mathcal{O}[f] \rangle \right)^2 \right]^{1/2} \qquad \text{uncertainty}$$

ILL-DEFINED PROBLEM

determine a set of infinite-dimensional objects (the FFs of each parton)
from a finite piece of information (the experimental data points)

2.2 Parton Parametrisation

Parametrisation: general features

Problem projected onto the finite-dimensional space of parameters

Choose a parametrisation at an initial scale Q_0^2
for each independent parton i (or a combination of them) and hadron h

$$\boxed{z f_i^h(z,Q_0^2) = A_{f_i^h}\, z^{a_{f_i^h}}\, (1-z)^{b_{f_i^h}}\, \mathscr{F}_i^h(z,\{c_{f_i^h}\})}$$

small z

$$z f_i^h(z,Q_0^2) \xrightarrow{z\to 0} z^{a_{f_i^h}}$$

$$\mathscr{F}_i^h(z,\{c_{f_i^h}\}) \xrightarrow[z\to 1]{z\to 0} \text{finite}$$

interpolation in between

large z

$$z f_i^h(x,Q_0^2) \xrightarrow{z\to 1} (1-z)^{b_{f_i^h}}$$

The problem is reduced to the determination of the finite set of parameters $\{c_{f_i^h}\}$

The interpolating function $\mathscr{F}_i^h(z,\{c_{f_i^h}\})$ should be sufficiently

GENERAL (the range of possible FF behaviours in the space of functions should not be limited)

SMOOTH (FFs are implicitly assumed ot be smooth functions)

FLEXIBLE (it should be able to adapt to a variety of data and processes)

to describe the data with minimal bias

Parametrisation: two alternative choices

1. Standard parametrisation, *e.g.*

$$\mathscr{F}_i^h(z,\{c_{f_i^h}\}) = \frac{1+\gamma_i^h(1-z)^{\delta_i^h}}{B[2+a_i^h,b_i^h+1]+\gamma_i^h B[2+a_i^h,b_i^h+\delta_i^h+1]}$$

in terms of a (relatively) small set of parameters ($\mathcal{O}(30)$ per PDF set)

$$\{\mathbf{a}\} = \{a_{f_i^h}, b_{f_i^h}, \gamma_{f_i^h}, \delta_{f_i^h}\}$$

$\Rightarrow$ **smooth behavior (a desirable feature for a PDF)**
$\Rightarrow$ **potential source of bias if the parametrization is too rigid**

2. Redundant parametrisation, *e.g.*

certain classes of polynomials (*e.g.* Bernstein, Chebyschev) or a neural network

in terms of a huge set of parameters ($\mathcal{O}(200)$ per PDF set)

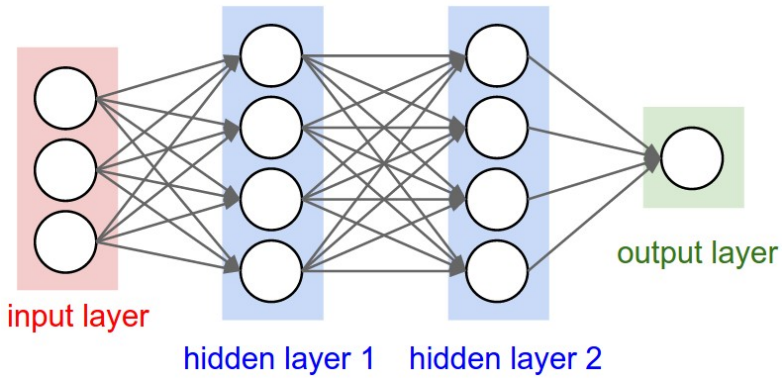
$$\{\mathbf{a}\} = \{\omega_{ij}^{(L-1),f_i^h}, \theta_i^{(L),f_i^h}\}$$

$\Rightarrow$ **potentially non-smooth**
$\Rightarrow$ **bias due to the parametrization reduced as much as possible**

Parametrisation: what a neural network exactly is?

A convenient functional form providing a flexible parametrization
used as a generator of random functions in the FF space

EXAMPLE: MULTY-LAYER FEED-FORWARD PERCEPTRON



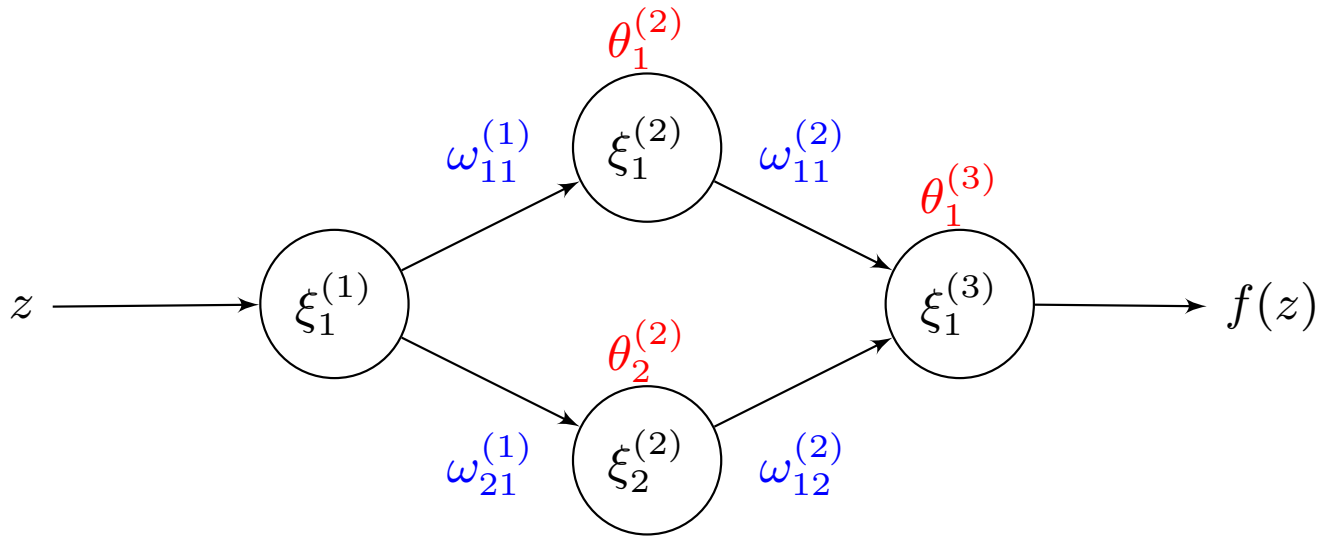
$$\xi_i^{(l)} = g\left(\sum_j^{n_l-1} \omega_{ij}^{(l-1)} \xi_j^{(l-1)} - \theta_i^{(l)}\right)$$

$$g(y) = \frac{1}{1+e^{-y}}$$

- made of neurons grouped into layers (define the architecture)
- each neuron receives input from neurons in the preceding layer (feed-forward NN)
- activation $\xi_i^{(l)}$ determined by a set of parameters (weights and thresholds)
- activation determined according to a non-linear function (except the last layer)

Parametrisation: what a neural network exactly is?

EXAMPLE: THE SIMPLEST 1-2-1 MULTI-LAYER FEED-FORWARD PERCEPTRON

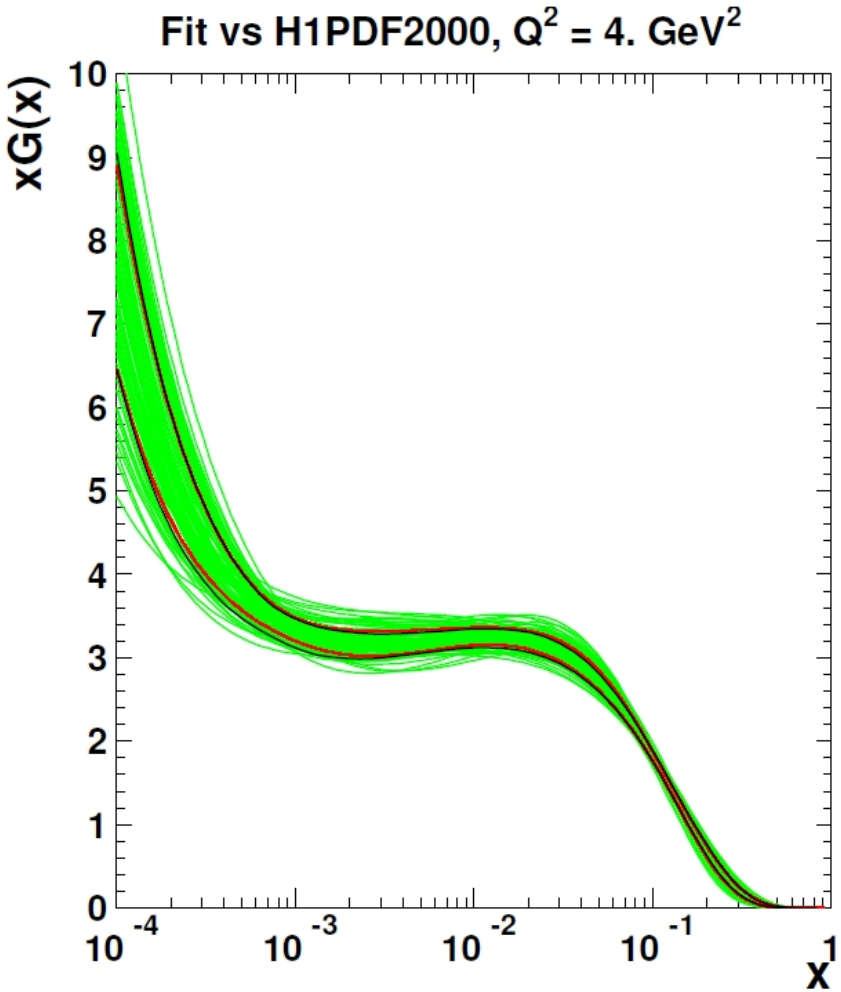


$$f(z) \equiv \xi_1^{(3)} = \left\{ 1 + \exp\left[\theta_1^{(3)} - \frac{\omega_{11}^{(2)}}{1 + e^{\theta_1^{(2)} - x\omega_{11}^{(1)}}} - \frac{\omega_{12}^{(2)}}{1 + e^{\theta_2^{(2)} - x\omega_{21}^{(1)}}} \right] \right\}^{-1}$$

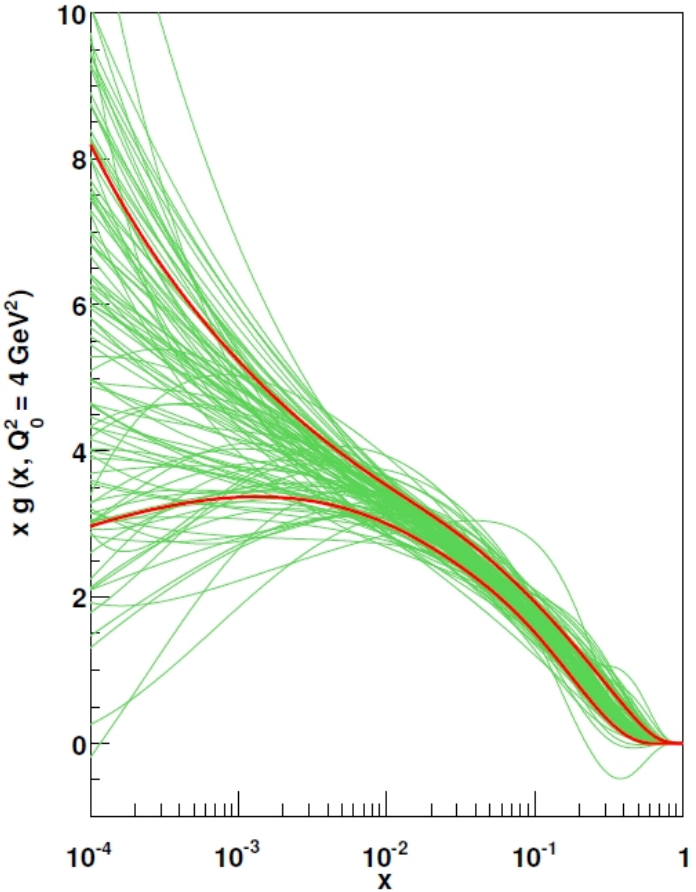
Recall: $$\xi_i^{(l)} = g\left(\sum_j^{n_l - 1} \omega_{ij}^{(l-1)} \xi_j^{(l-1)} - \theta_i^{(l)} \right) \; ; \qquad g(z) = \frac{1}{1 + e^{-z}}$$

Parametrisation: standard *vs* redundant

HERA-LHC 2009 PDF benchmark



simple parametrization



redundant parametrization (NN)

2.3 Goodness-of-fit

Fit quality

1. Define the fit quality (the χ^2 function)

$$\chi^2 = \sum_{i,j}^{\mathrm{N_{dat}}} \left(T_i[\{\mathbf{a}\}] - D_i\right) \left(\mathrm{cov}^{-1}\right)_{ij} \left(T_j[\{\mathbf{a}\}] - D_j\right)$$

with the experimental covariance matrix

$$\left(\mathrm{cov}\right)_{ij} = \delta_{ij} s_i^2 + \left(\sum_{\alpha}^{N_c} \sigma_{i,\alpha}^{(c)} \sigma_{j,\alpha}^{(c)} + \sum_{\alpha}^{N_{\mathcal{L}}} \sigma_{i,\alpha}^{(\mathcal{L})} \sigma_{j,\alpha}^{(\mathcal{L})} \right) D_i D_j$$

s_i are N_{dat} uncorrelated uncertainties (statistic + uncorrealted systematic ucnertainties)
$\sigma_{i,\alpha}^{(c)}$ are $N_{\mathrm{dat}} \times N_c$ additive correlated uncertainties
$\sigma_{i,\alpha}^{(\mathcal{L})}$ are $N_{\mathrm{dat}} \times N_{\mathcal{L}}$ multiplicative uncertainties

2. Find the best-fit configuration of parameters $\{\mathbf{a_0}\}$ which minimise the χ^2
3. Treat conveniently
   - uncorrelated/correlated uncertainties
     need not to overestimate uncertainties and to let the χ^2 be faithful
   - additive/multiplicative uncertainties
     need to avoid the D'Agostini bias

Normalisation uncertainties: D'Agostini bias [JHEP 1005 (2010) 075]

1. Consider one experiment with $N_{\rm dat}$ data d_i of one theoretical quantity t

$$\chi^2(t) = \sum_{i,j}^{N_{\rm dat}} (t - d_i) \left({\rm cov}^{-1}\right)_{ij} (t - d_j)$$

2. The best-fit theoretical quantity t_0 and its variance v_t are given by

$$\left.\frac{d\chi^2}{dt}\right|_{t=t_0} = 0 \Longleftrightarrow t_0 = \frac{\sum_{i,j}^{N_{\rm dat}} \left({\rm cov}^{-1}\right)_{ij} d_j}{\sum_{i,j}^{N_{\rm dat}} \left({\rm cov}^{-1}\right)_{ij}} \qquad v_t = \left(\frac{1}{2}\frac{d^2\chi^2}{dt^2}\right)^{-1} = \frac{1}{\sum_{i,j}^{N_{\rm dat}} \left({\rm cov}^{-1}\right)_{ij}}$$

3. Consider completely uncorrelated additive errors: $({\rm cov})_{ij} = s_i^2 \delta_{ij}$

$$t_0 = w = \Sigma^2 \sum_i^{N_{\rm dat}} \frac{d_i}{s_i^2} \qquad v_t = \Sigma^2 \qquad \text{with } \frac{1}{\Sigma^2} = \sum_i^{N_{\rm dat}} \frac{1}{s_i^2}$$

4. Consider an additional common normalisation error: $({\rm cov})_{ij} = (s_i^2 + \sigma^2 d_i^2)\delta_{ij}$

$$t_0 = \frac{w}{1 + r^2\sigma^2 w^2/\Sigma^2} \qquad v_t = \frac{\Sigma^2 + \sigma^2 w^2 (1 + r^2)}{1 + r^2\sigma^2 w^2/\Sigma^2} \qquad \text{with } r^2 = \frac{\Sigma^2}{w^2} \sum_i^{N_{\rm dat}} \frac{(d_i - w)^2}{s_i^2}$$

5. Both t_0 and v_t are affected by a downward bias
   smaller values of d_i have a smaller normalization uncertainties σd_i and are thus preferred

Normalisation uncertainties: D'Agostini bias [JHEP 1005 (2010) 075]

1. The penalty trick: redefine the fit quality

$$\chi^2(t) \to \chi^2(t,\mathcal{N}) = \sum_i^{N_{\rm dat}} \frac{(t/\mathcal{N}-d_i)^2}{s_i^2} + \frac{(\mathcal{N}-1)^2}{\sigma^2}$$

$$\left.\frac{\partial\chi^2}{\partial t}\right|_{t=t_0} = \frac{\partial\chi^2}{\partial\mathcal{N}} = 0 \Longleftrightarrow t_0 = w \qquad v_t = \left(\frac{1}{2}\frac{d^2\chi^2}{dt^2}\right)^{-1} = \Sigma^2 + \sigma^2 w^2$$

$\longrightarrow$ recover the unbiased estimators for t_0 and v_t

2. The t_0 method: redefine the covariance matrix

$$(\mathrm{cov})_{ij} \to (\mathrm{cov}_{t_0})_{ij} \Longleftrightarrow (s_i^2 + \sigma^2 d_i^2)\delta_{ij} \to s_i^2\delta_{ij} + \sigma^2 t_0^2$$

$$\left(\mathrm{cov}_{t_0}^{-1}\right)_{ij} = \frac{\delta_{ij}}{s_i^2} - \frac{\sigma^2 t_0^2}{s_i^2 s_j^2}\frac{\Sigma^2}{\Sigma^2 + \sigma^2 t_0^2} \Longleftrightarrow t_0 = w \qquad v_t = \Sigma^2 + \sigma^2 w^2$$

$\longrightarrow$ recover the unbiased estimators for t_0 and v_t, provided that w is tuned to t_0
$\longrightarrow$ w can be tuned to t_0 via an iterative procedure

The d'Agostini bias and its solution can be generalised to more than one experiment with different normalisation errors (per experiment/per data point)

2.4 Parameter optimisation

Parameter optimisation: general framework

Optimisation usually performed by means of simple gradient descent:
compute and minimise the gradient of the fit quality with respect to the fit parameters

$$\frac{\partial \chi^2}{\partial a_i}, \qquad \text{for } i = 1, \ldots, N_{\rm par}$$

Optimisation should minimise the noise in the χ^2 driven by noisy experimental data

Additional complications in case of a redundant parametrisation (huge parameter space)

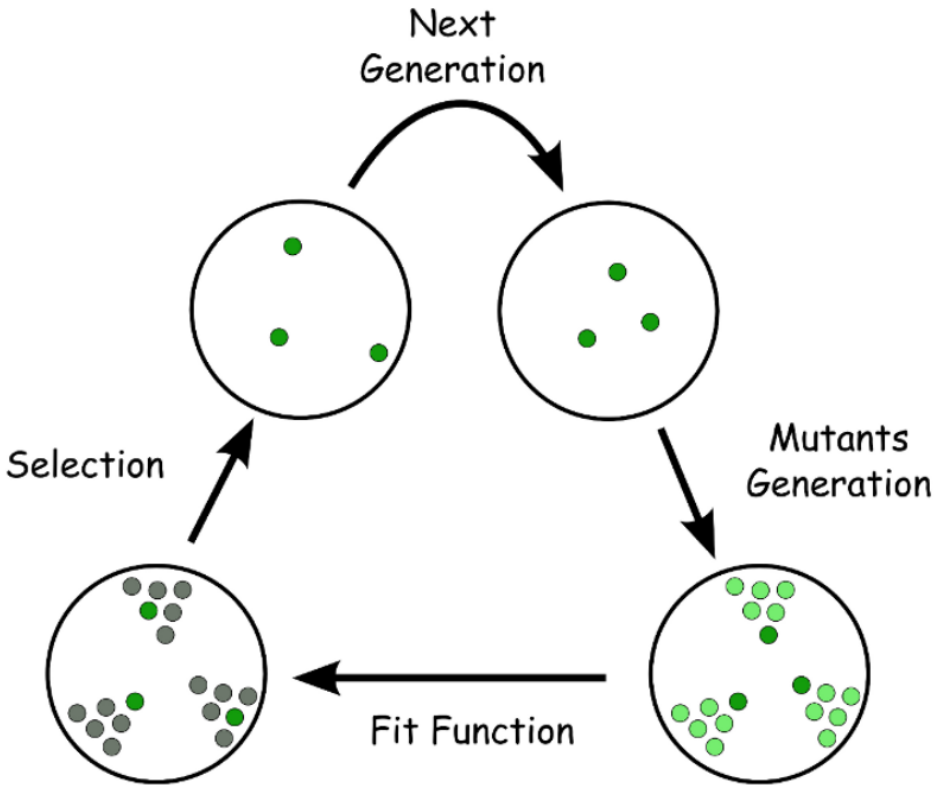
1. need to explore the parameter space as uniformly as possible
   (in order to avoid stopping the fit in a local minimum)
2. need for a computationally efficient minimisation
   (non-trivial relationship between FFs and observables via convolution)
3. need to define a criterion for minimisation stopping
   (avoid learning statistical fluctuations of the data)

Alternative algorithms:
genetic algorithms, adaptive algorithms, ...

Minimisation: genetic algorithm

1. Initial population of NNs
   $\longrightarrow$ pars initialised randomly
2. Mutants generation
   $\longrightarrow$ mutations are introduced
3. Fit function
   $\longrightarrow$ the total χ^2 is computed
4. Selection
   $\longrightarrow$ best pars configs selected
5. Next generation
   $\longrightarrow$ iterate the process

until convergence is achieved



Good exploration of the parameter space

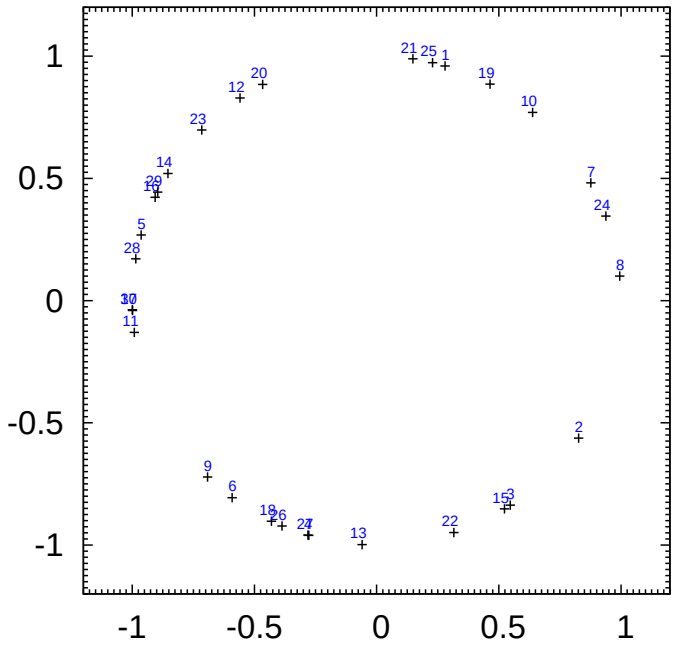
No need to compute gradients

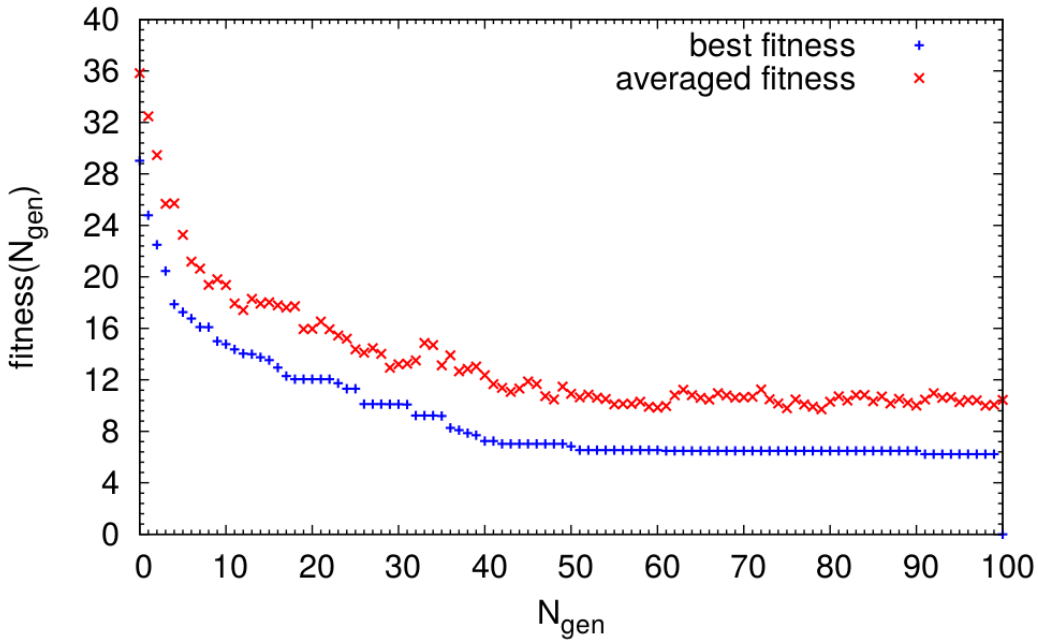
Lower computational efficiency than standard gradient gradient descent

Possible sensitivity to noise in χ^2 driven by noisy data

Example: the traveling salesman problem

Determine the shortest path among 30 towns on a circumference
(the fitness, or fit quality, is the distance connecting all 30 points)

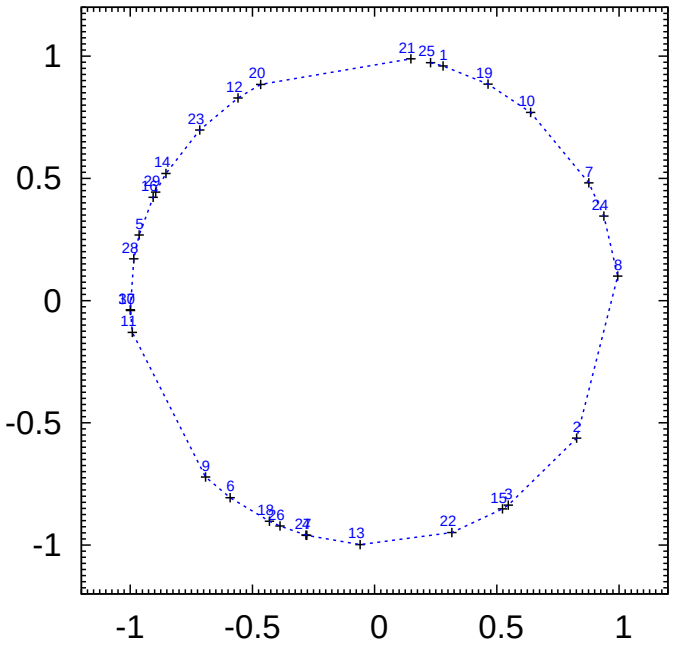


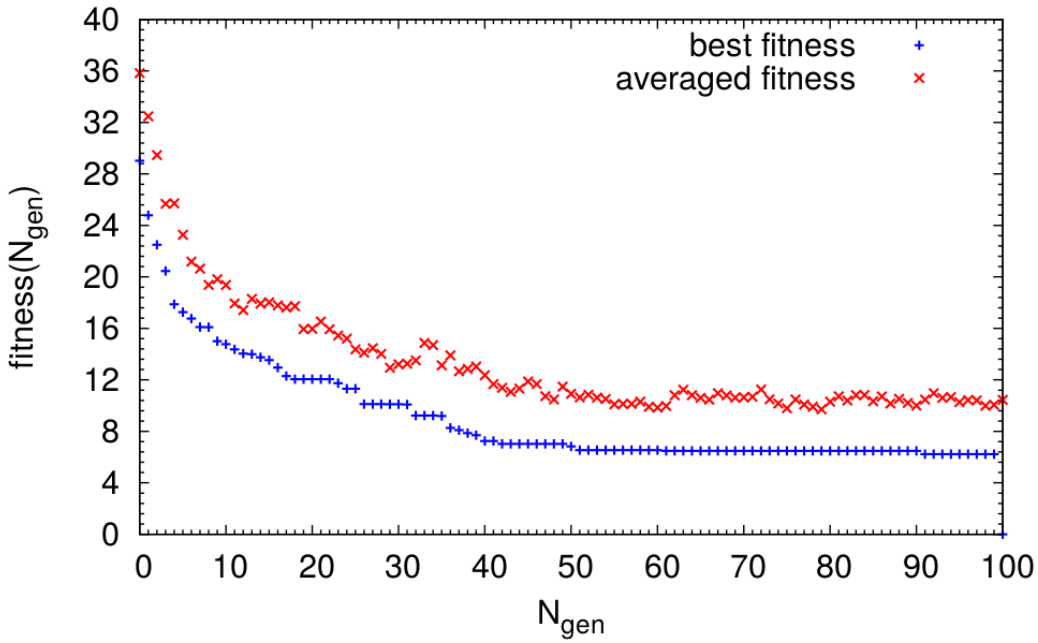



```
Initial population
  best fitness     29.0239
    2   17    3    9   18   15    6   23   28   12   16   29   14   27   26   30    7    8   24   20   21   22   10   25    1   19    5    4   11   13

    Generation   1
  best fitness     24.7844
   17    6   23   29   14   25    1   21   22   18   15   20   10    7    8   24   19    3    9   27   26    2   28   16   30   12    5   13    4   11

    Generation   2
  best fitness     22.4962
   12   16   30   28    5   17    6   23   29   14   25    1   21    8   24   22   18   27   26    2   13   15   20   10    7   19    3    4   11    9

    Generation   3
  best fitness     20.4627
   14   25    1   21   19   24   22   18   27   26    2   13   15   20   10    7    8    3    4   11    9   17   28    5   30    6   23   29   12   16
```

Example: the traveling salesman problem

Determine the shortest path among 30 towns on a circumference
(the fitness, or fit quality, is the distance connecting all 30 points)

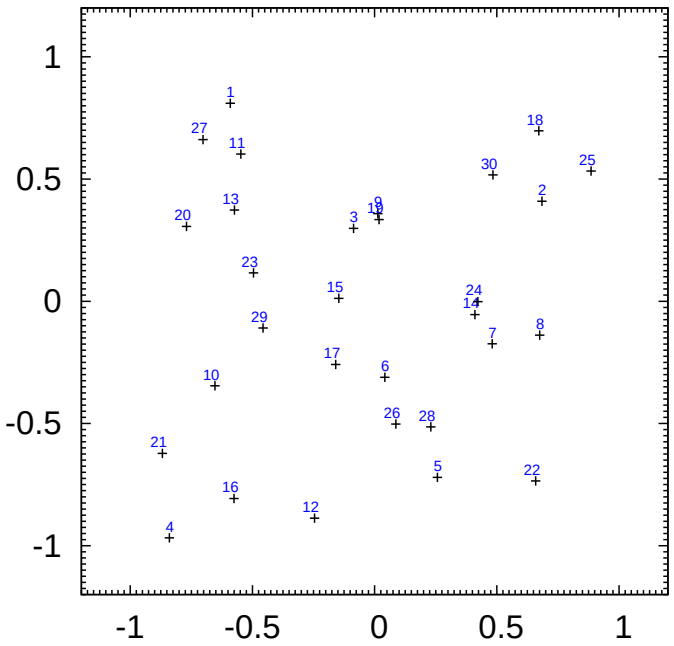


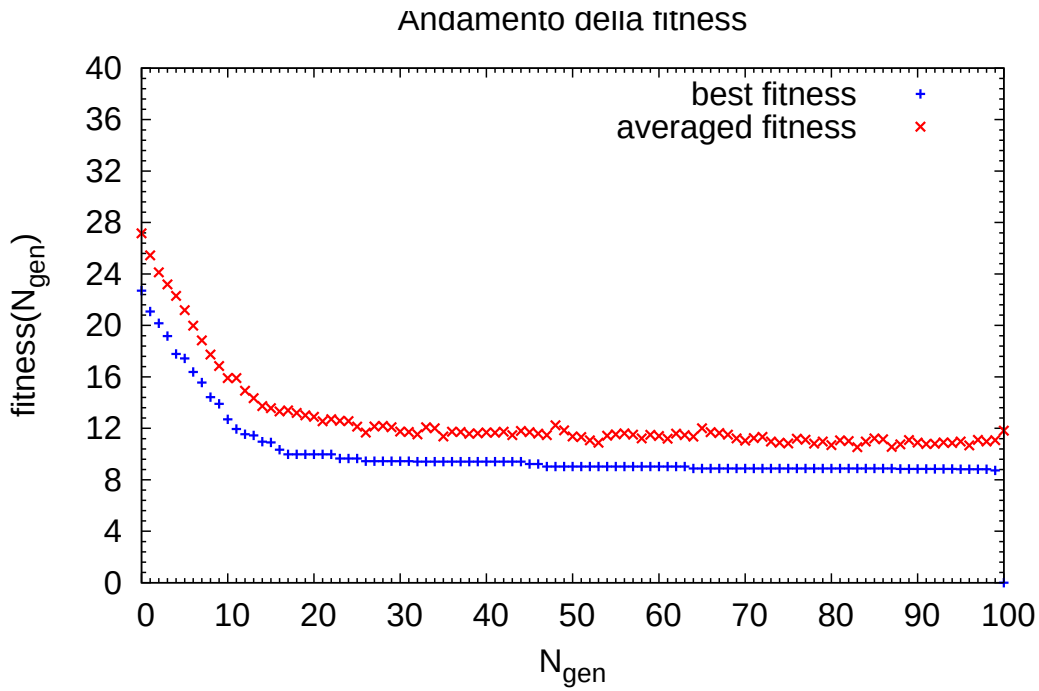



```
Initial population
  best fitness     29.0239
    2   17    3    9   18   15    6   23   28   12   16   29   14   27   26   30    7    8   24   20   21   22   10   25    1   19    5    4   11   13

    Generation   1
  best fitness     24.7844
   17    6   23   29   14   25    1   21   22   18   15   20   10    7    8   24   19    3    9   27   26    2   28   16   30   12    5   13    4   11

    Generation   2
  best fitness     22.4962
   12   16   30   28    5   17    6   23   29   14   25    1   21    8   24   22   18   27   26    2   13   15   20   10    7   19    3    4   11    9

    Generation   3
  best fitness     20.4627
   14   25    1   21   19   24   22   18   27   26    2   13   15   20   10    7    8    3    4   11    9   17   28    5   30    6   23   29   12   16
```

Example: the traveling salesman problem

Determine the shortest path among 30 towns in a square
(the fitness, or fit quality, is the distance connecting all 30 points)

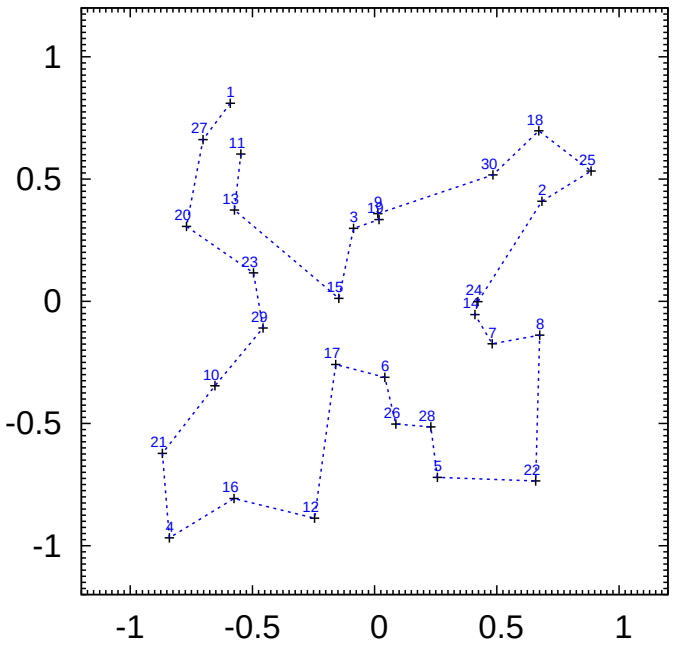


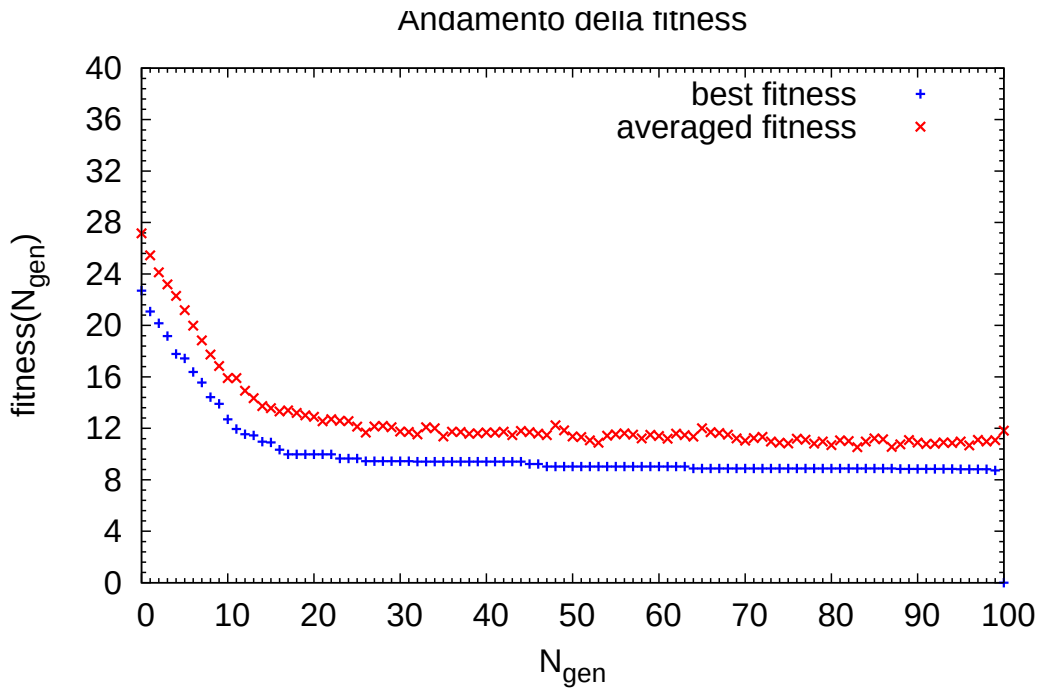



```
Initial population
  best fitness     22.704
   12    5   19    9    3    8   22    7   25   18   15   11   30   16   26   21   23   10   13    1   20   24   14   29    2   28    6   27   17    4

    Generation   1
  best fitness    21.0794
   16   13    1   19    9   29    2   25   18   28    6   10   20   11   30   17    4   24   14   27    3   23   15    5    8    7   26   22   21   12

    Generation   2
  best fitness      20.17
    6   10   13    1   19    9   20   11   23   15   24   14   22   21   16   29    2   25   18   28   27    3    8    7   30   17    4    5   26   12

    Generation   3
  best fitness    19.1733
    3    8   14   22   28   27    4    5   26   12   21   16    6   10   13    1   19    9   20   11   23   15   17   30    7    2   18   25   24   29
```

Example: the traveling salesman problem

Determine the shortest path among 30 towns in a square
(the fitness, or fit quality, is the distance connecting all 30 points)






```
Initial population
  best fitness     22.704
   12    5   19    9    3    8   22    7   25   18   15   11   30   16   26   21   23   10   13    1   20   24   14   29    2   28    6   27   17    4

    Generation   1
  best fitness    21.0794
   16   13    1   19    9   29    2   25   18   28    6   10   20   11   30   17    4   24   14   27    3   23   15    5    8    7   26   22   21   12

    Generation   2
  best fitness      20.17
    6   10   13    1   19    9   20   11   23   15   24   14   22   21   16   29    2   25   18   28   27    3    8    7   30   17    4    5   26   12

    Generation   3
  best fitness    19.1733
    3    8   14   22   28   27    4    5   26   12   21   16    6   10   13    1   19    9   20   11   23   15   17   30    7    2   18   25   24   29
```

Minimisation: adaptive algorithms [N. Hansen, Springer (2016)]

The Covariance Matrix Adaption - Evolution Strategy (CMA-ES)

1. Initialisation at the (0)-th generation

$$\mathbf{a}^{(0)} \sim \mathcal{N}(0, \mathbf{C}^{(0)}) \,, \qquad \mathbf{C}^{(0)} = \mathbf{I}$$

2. Mutation at the (i)-th generation, λ mutants, step-size $\sigma^{(i-1)}$

$$\mathbf{x}_k^{(i)} \sim \mathbf{a}^{(i-1)} + \sigma^{(i-1)} \mathcal{N}(0, \mathbf{C}^{(i-1)}) \,, \qquad \text{for } k = 1, \ldots, \lambda$$

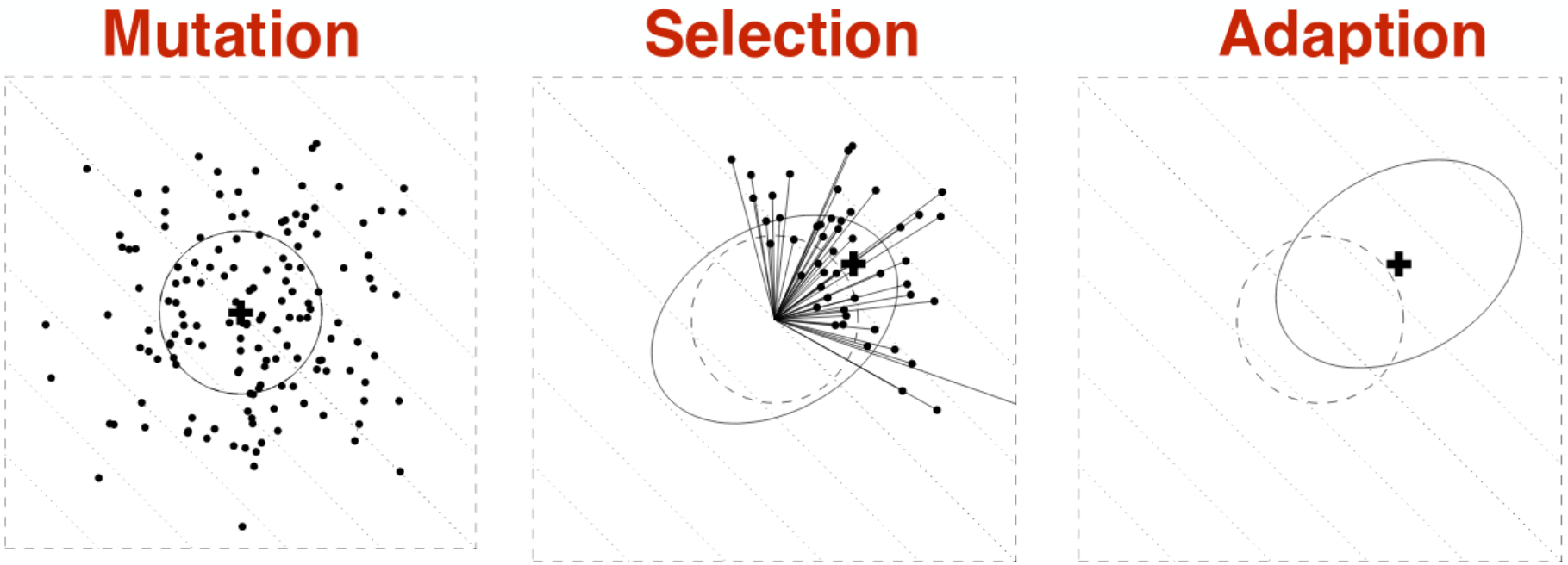
   compute the fitness of each mutant and rank them such that $\chi^2(\mathbf{x}_k) < \chi^2(\mathbf{x}_{k+1})$

3. (Non-elitist) recombination
   compute the new search centre as a weighted average over the $\mu = \lambda/2$ best mutants

$$\mathbf{a}^{(i)} = \mathbf{a}^{(i-1)} + \sum_{i=1}^{\mu} w_i \left(\mathbf{x}_k^{(i)} - \mathbf{a}^{(i-1)} \right)$$

   update $\mathbf{C}$ using information on the parameter space learnt from the mutants
   iterate until convergence is reached

Minimisation: the CMA-ES algorithms [N. Hansen, Springer (2016)]



The key features of the CMA-ES family of algorithms are
the determination of the search distribution covariance matrix $\mathbf{C}^{(i)}$ (and possibly of the step-size σ^i)

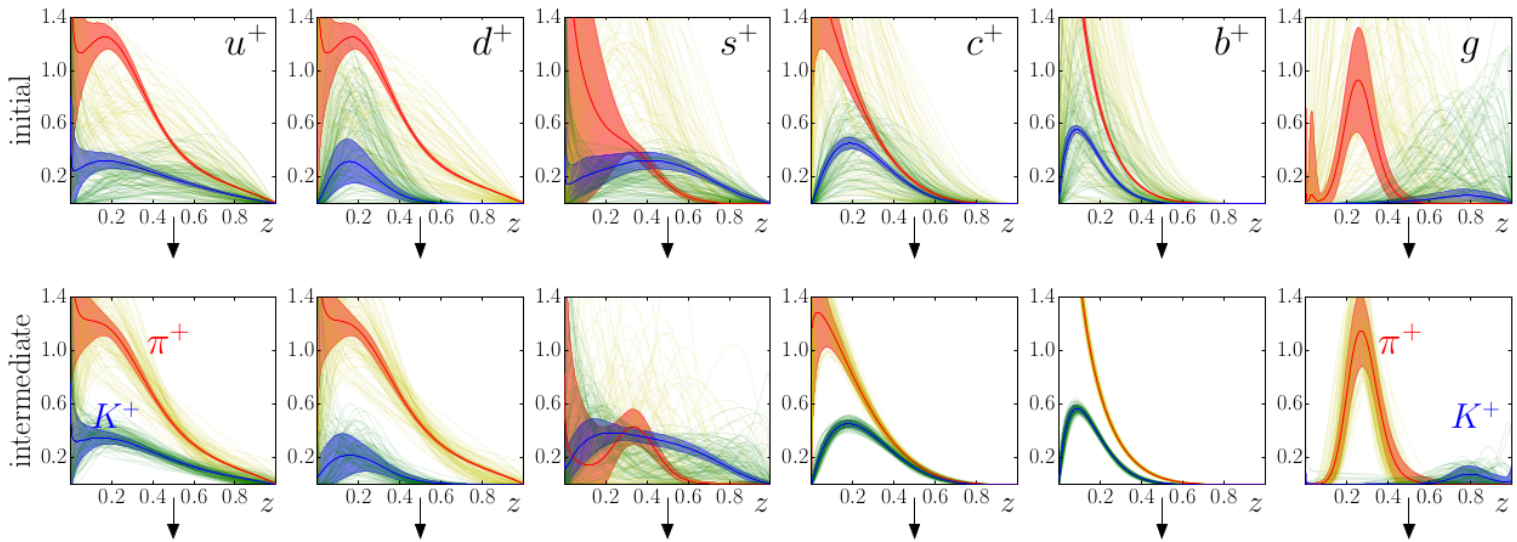
These features are optimised by the fit procedure,
making use of the information present in the ensemble of mutants
to learn preferred directions in parameter space

Internal parameters ($\sigma^{(0)}$, λ, w_i) tuned by trial and error

The iterative Monte Carlo procedure [PRD 93 (2016) 074005, PRD 94 (2016) 114004]

1. Generate a *prior* set of parameters $\{\mathbf{a}\}^{(0)}$
   using a flat sampling in the parameter space
2. Minimise the χ^2 and determine the *posterior* set of best-fit parameters $\{\mathbf{a_0}\}^{(0)}$
3. Generate a new prior set of parameters $\{\mathbf{a}\}^{(1)}$
   using a multivariate Gaussian sampling in the parameter space
   with covariance matrix and central parameters determined from $\{\mathbf{a_0}\}^{(0)}$
4. Minimise the χ^2 and determine the *posterior* set of best-fit parameters $\{\mathbf{a_0}\}^{(1)}$
5. Iterate until convergence is reached

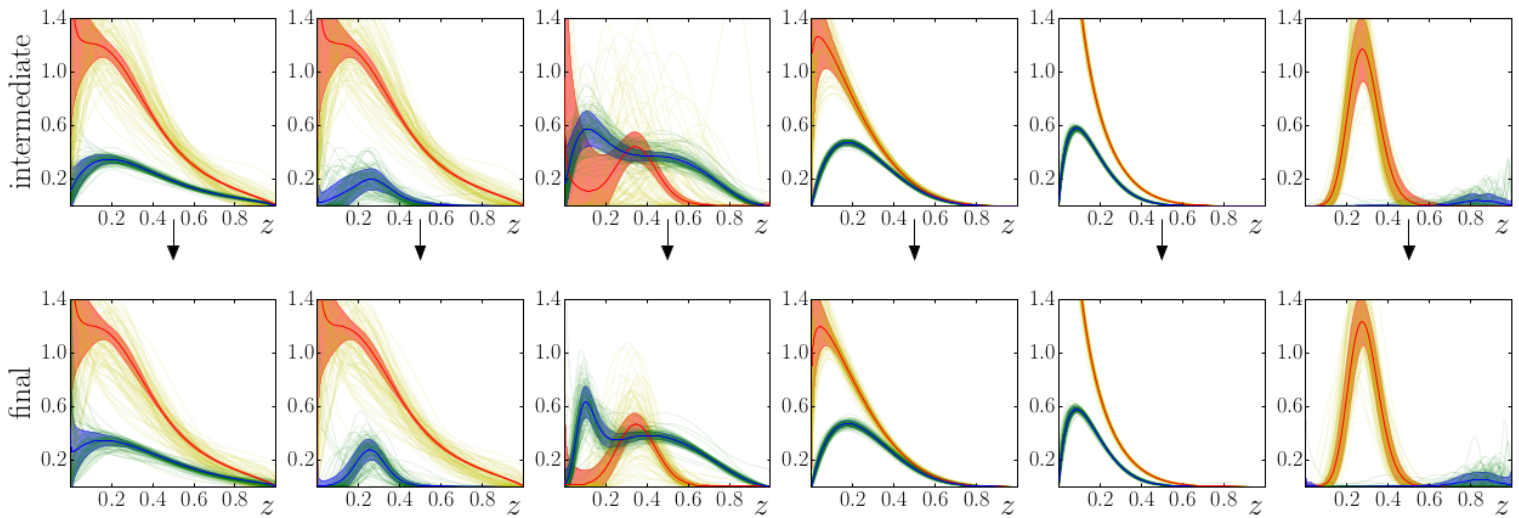
Efficient exploration of the parameter space (avoid local minima)



The iterative Monte Carlo procedure [PRD 93 (2016) 074005, PRD 94 (2016) 114004]

1. Generate a *prior* set of parameters $\{\mathbf{a}\}^{(0)}$
   using a flat sampling in the parameter space
2. Minimise the χ^2 and determine the *posterior* set of best-fit parameters $\{\mathbf{a_0}\}^{(0)}$
3. Generate a new prior set of parameters $\{\mathbf{a}\}^{(1)}$
   using a multivariate Gaussian sampling in the parameter space
   with covariance matrix and central parameters determined from $\{\mathbf{a_0}\}^{(0)}$
4. Minimise the χ^2 and determine the *posterior* set of best-fit parameters $\{\mathbf{a_0}\}^{(1)}$
5. Iterate until convergence is reached

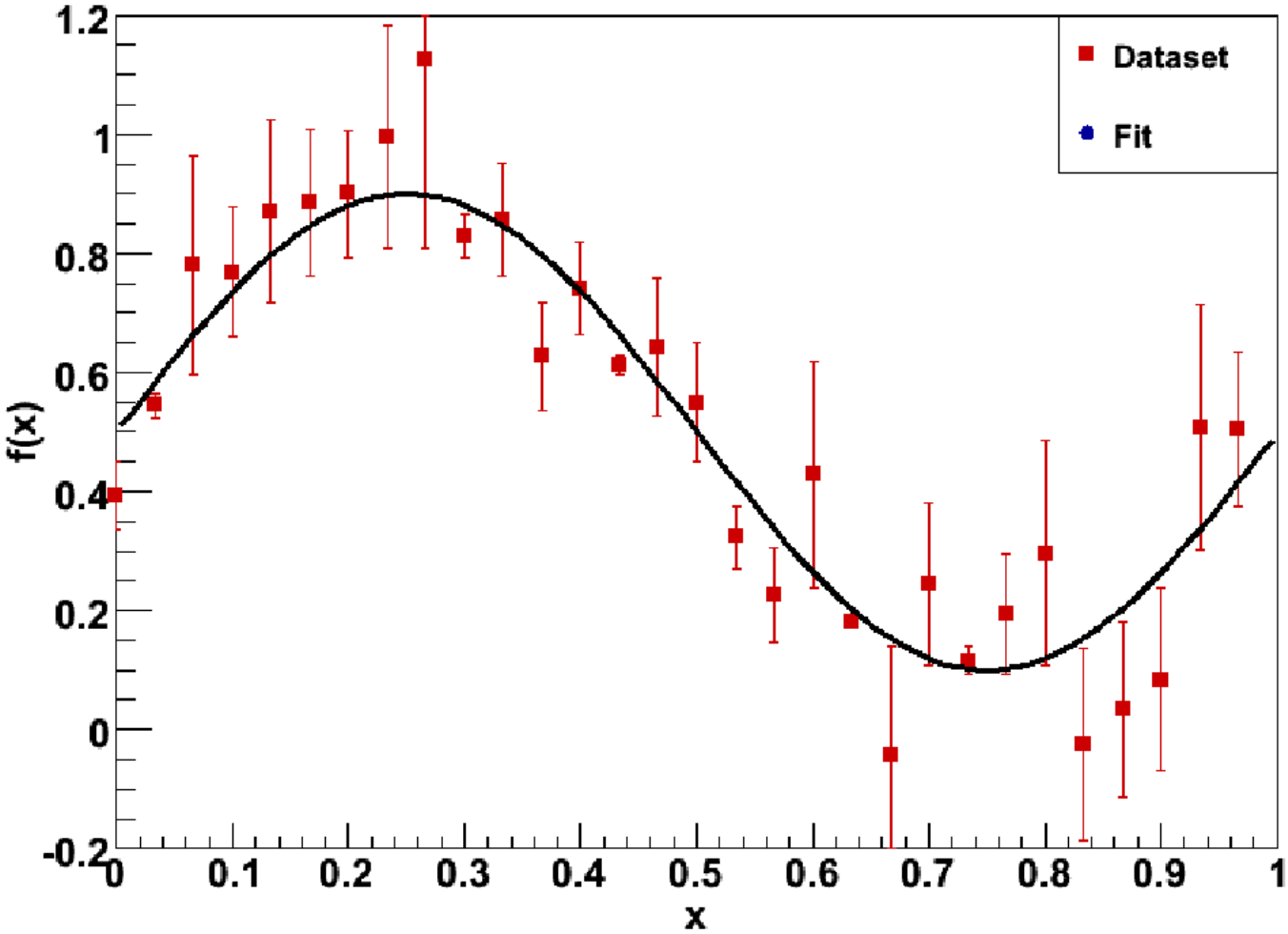
Efficient exploration of the parameter space (avoid local minima)



Minimisation: learning

If the parametrisation is redundant, statistical noise in the data can be learnt

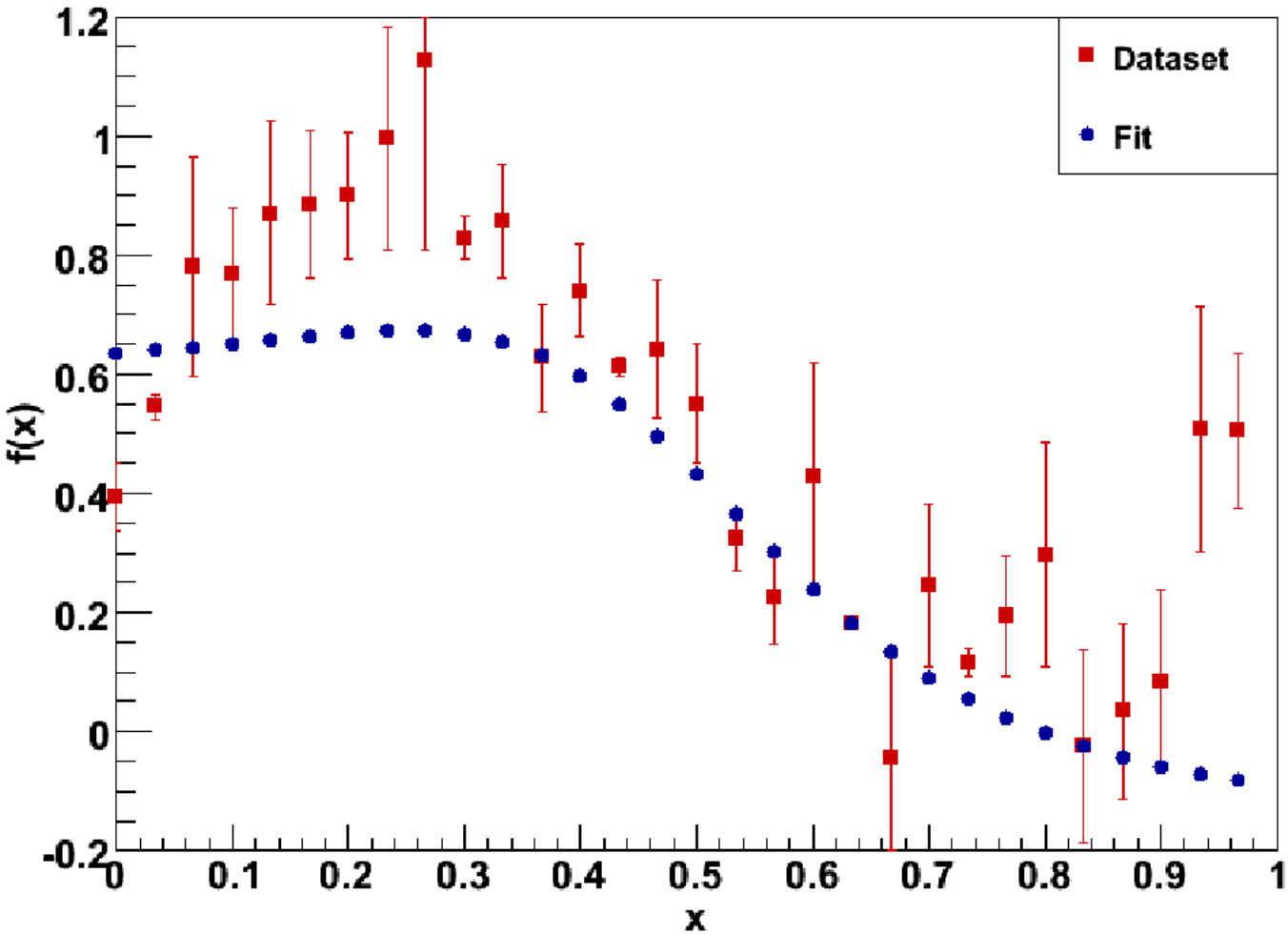
UNDERLYING PHYSICAL LAW



Minimisation: learning

If the parametrisation is redundant, statistical noise in the data can be learnt

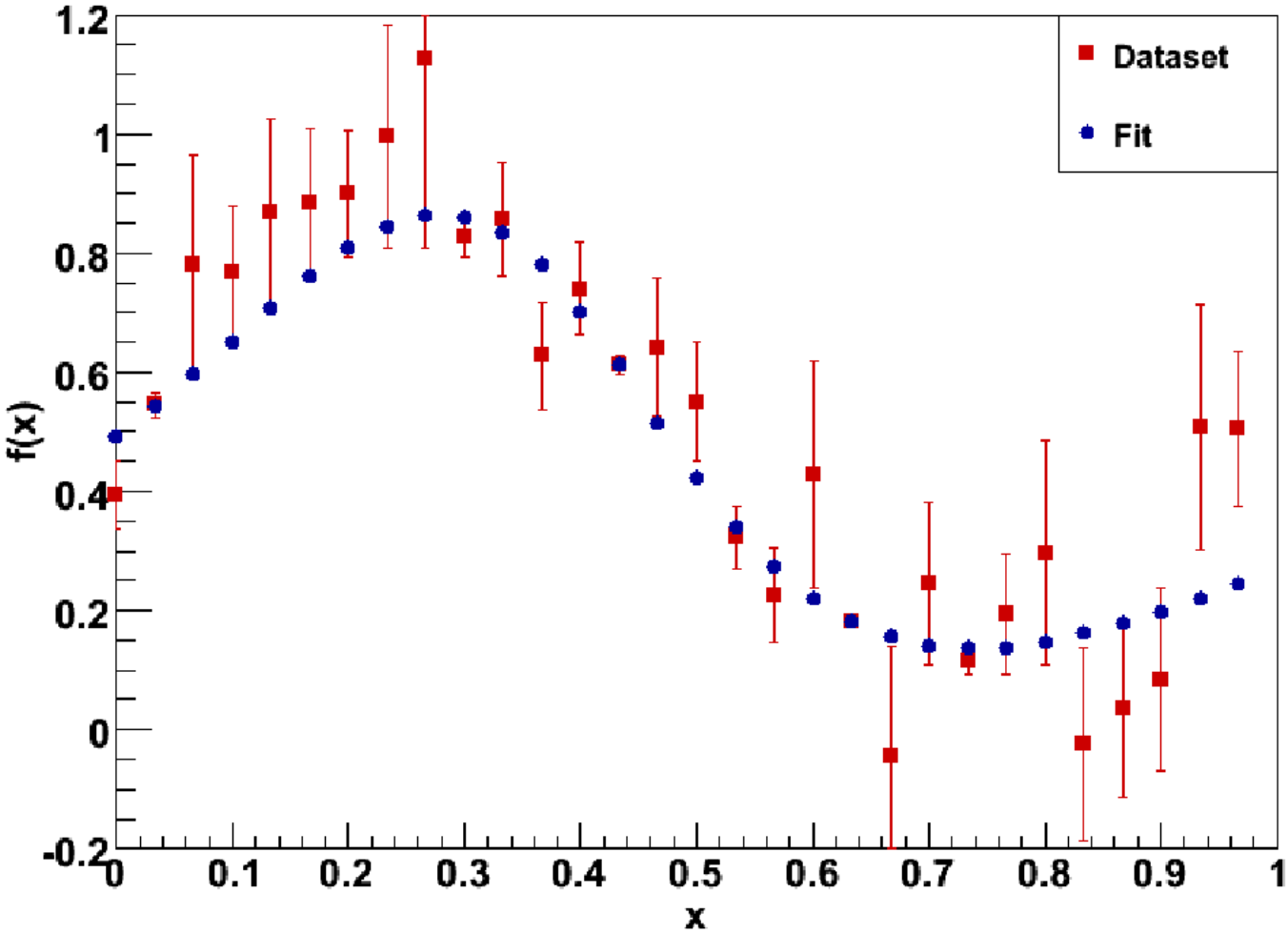
UNDERLEARNING



Minimisation: learning

If the parametrisation is redundant, statistical noise in the data can be learnt

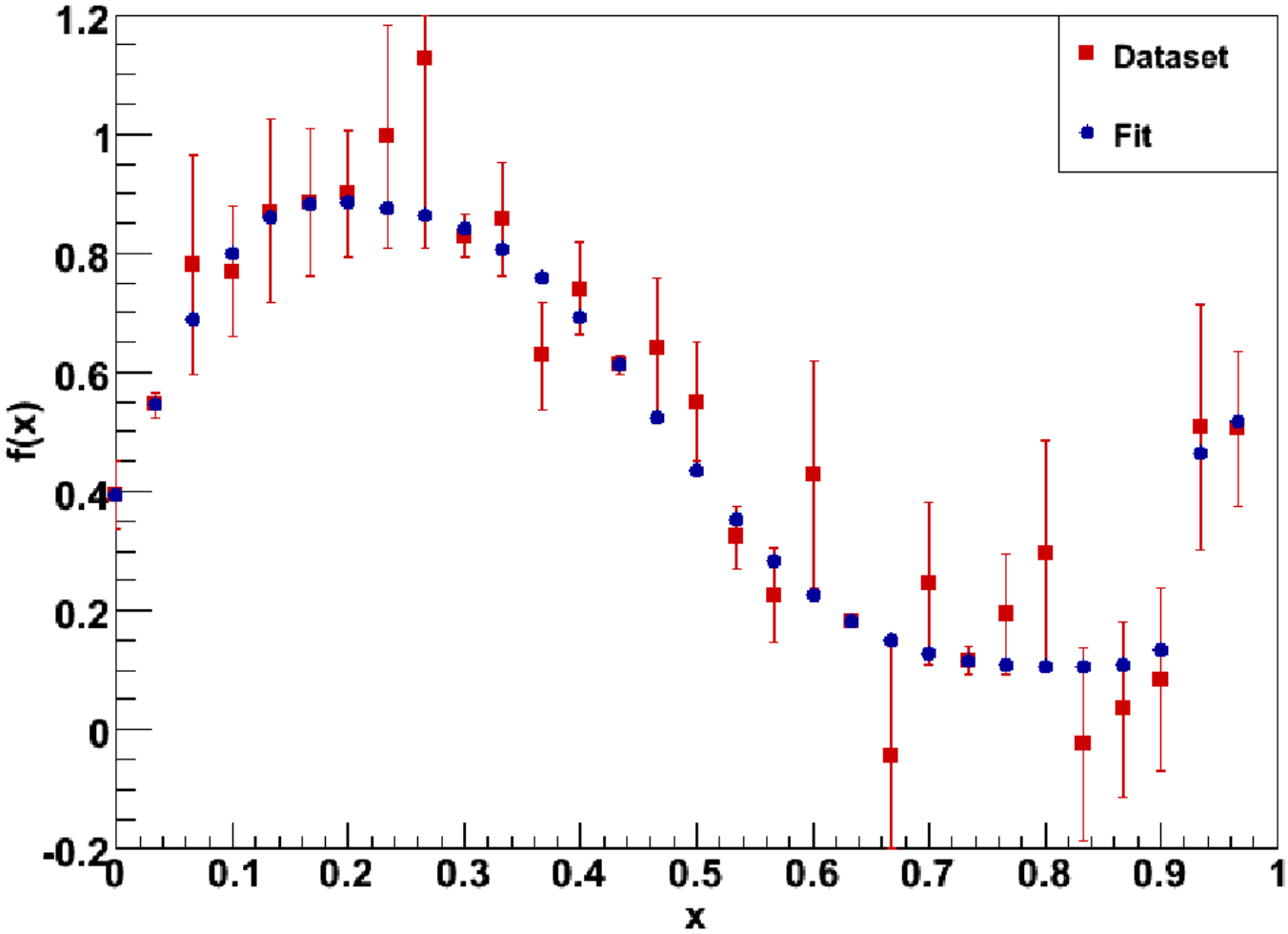
PROPER LEARNING



Minimisation: learning

If the parametrisation is redundant, statistical noise in the data can be learnt

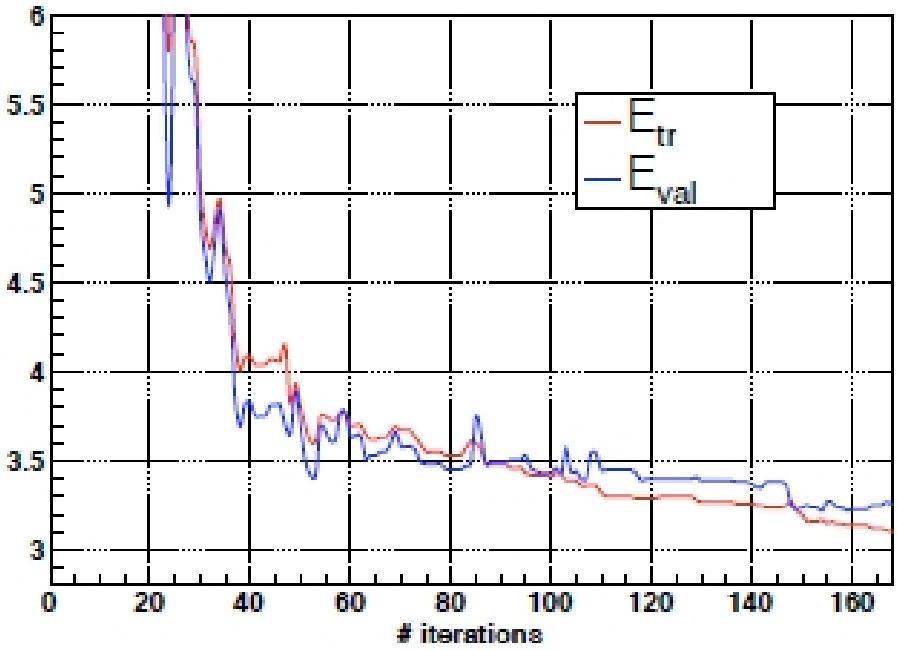
OVERLEARNING

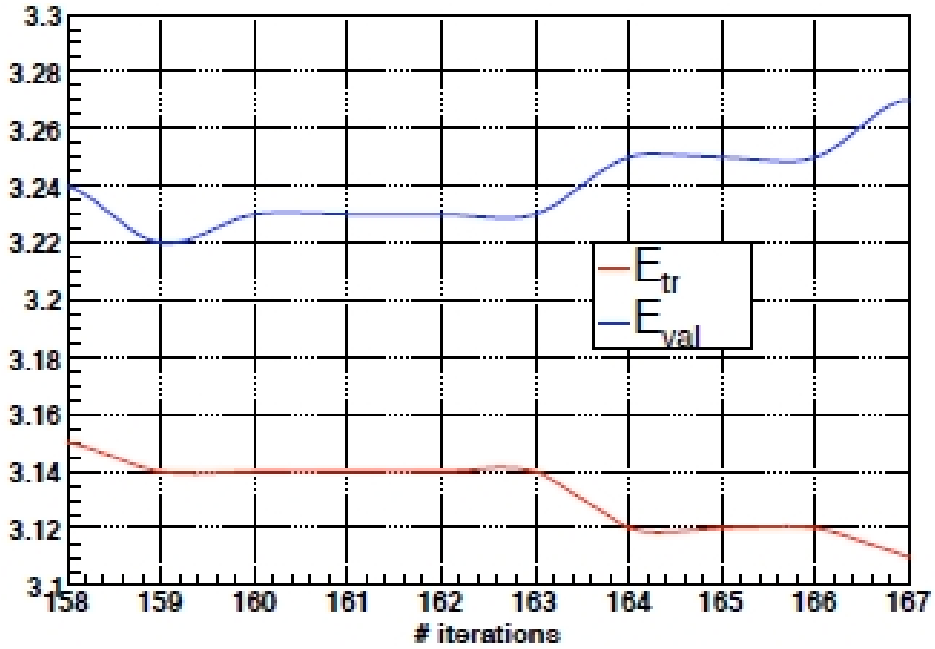


Minimisation: stopping criterion

CROSS-VALIDATION METHOD

- divide the data into two subsets (training & validation)
- train the NN on training subset and compute χ^2 for each subset
- stop when the χ^2 of validation subset no longer decreases (NN are learning noise!)





The best fit does not coincide with the χ^2 absolute minimum

2.5 Representation of uncertainties

The Hessian method: general strategy

1. Expand the χ^2 about its global minimum at first (nontrivial) order

$$\chi^2\{\mathbf{a}\} \approx \chi^2\{\mathbf{a_0}\} + \delta a^i H_{ij} \delta a^j, \qquad H_{ij} = \frac{1}{2} \left. \frac{\partial^2 \chi^2\{\mathbf{a}\}}{\partial a_i \partial a_j} \right|_{\{\mathbf{a}\}=\{\mathbf{a_0}\}}$$

2. Assume linear error propagation for any observable $\mathcal{O}$ depending on $\{\mathbf{a}\}$

$$\langle \mathcal{O}\{\mathbf{a}\} \rangle \approx \mathcal{O}\{\mathbf{a_0}\} + a_i \left. \frac{\partial \mathcal{O}\{\mathbf{a}\}}{\partial a_i} \right|_{\{\mathbf{a}\}=\{\mathbf{a_0}\}} \qquad \sigma_{\mathcal{O}\{\mathbf{a}\}} \approx \sigma_{ij} \frac{\partial \mathcal{O}\{\mathbf{a}\}}{\partial a_i} \left. \frac{\partial \mathcal{O}\{\mathbf{a}\}}{\partial a_j} \right|_{\{\mathbf{a}\}=\{\mathbf{a_0}\}}$$

3. Determine σ_{ij} from H_{ij} from maximum likelihood (under Gaussian hypothesis)

$$\sigma_{ij}^{-1} = \left. \frac{\partial^2 \chi^2\{\mathbf{a}\}}{\partial a_i \partial a_j} \right|_{\{\mathbf{a}\}=\{\mathbf{a_0}\}} = H_{ij}$$

4. A C.L. about the best fit is obtained as the volume (in parameter space) about $\chi^2\{\mathbf{a_0}\}$ that corresponds to a fixed increase of the χ^2; for Gaussian uncertainties:

$$68\% \text{ C.L.} \Longleftrightarrow \Delta\chi^2 = \chi^2\{\mathbf{a}\} - \chi^2\{\mathbf{a_0}\} = 1$$

The Hessian method: some remarks

1. Compact representation and computation of observables and their uncertainties

$$\langle \mathcal{O}[f(x,Q^2)] \rangle = \mathcal{O}[f_0(x,Q^2)]$$

$$\sigma_{\mathcal{O}}[f(x,Q^2)] = \frac{1}{2} \left[\sum_{i=1}^{N_{\rm par}} \left(\mathcal{O}[f_i(x,Q^2)] - \mathcal{O}[f_0(x,Q^2)] \right)^2 \right]^{1/2}$$

2. Parameters can always be adjusted so that all eigenvalues of H_{ij} are equal to one (diagonalise H_{ij} and rescale the eigenvectors by their eigenvalues)

$$\delta a_i H_{ij} \delta a_j = \sum_{i=1}^{N_{\rm par}} \left[a_i'(a_i) \right]^2 \Longleftrightarrow \sigma_{\mathcal{O}\{\mathbf{a}'\}} = \left| \nabla' \mathcal{O}\{\mathbf{a}'\} \right|$$

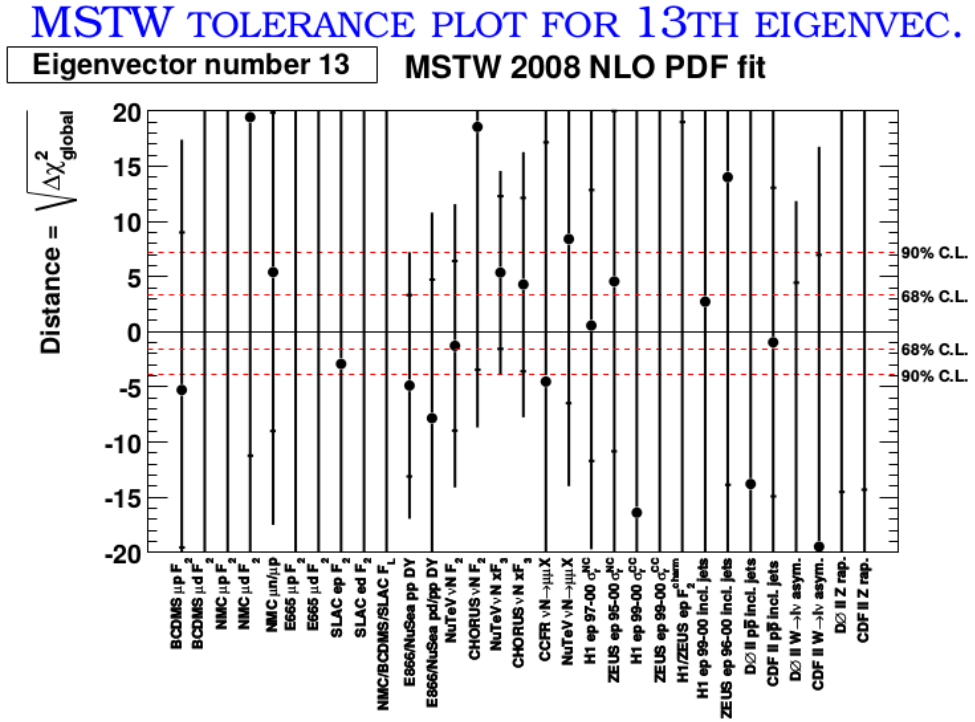
   The total contribution to the uncertainty due to two different sources (possibly correlated) is obtained by simply adding them in quadrature

3. Any rotation in the space of parameters preserves the gradient (one can diagonalise a chosen observable without spoiling the result)

4. Unmanageable Hessian matrix if the numer of parameters is huge

The Hessian method: limitations

Uncertainties obtained with $\Delta\chi^2 = 1$ might be unrealistically small
(inadequacy of the linear approximation)



uncertainties tuned to the distribution of deviations from best-fits for single experiments

for each eigenvector in parameter space
determine the CL for the distribution of best-fits of each experiment

rescale to the $\Delta\chi^2 = T$ interval such that correct confidence intervals are reproduced

no statistically rigorous interpretation of T (tolerance)

The Monte Carlo method: general strategy

1. Generate (art) replicas of (exp) data according to the distribution

$$\mathcal{O}_i^{(art)(k)} = \mathcal{O}_i^{(exp)} + r_i^{(k)} \sigma_{\mathcal{O}_i} \,, \qquad i = 1, \ldots N_{\rm dat} \,, \qquad k = 1, \ldots, N_{\rm rep}$$

   where $r_i^{(k)}$ are (Gaussianly distributed) random numbers for each k-th replica ($r_i^{(k)}$ can be generated with any distribution, not neccesarily Gaussian)

2. Perform a fit for each replica $k = 1, \ldots, N_{\rm rep}$

3. Compact computation of observables and their uncertainties (PDF replicas are equally probable members of a statistical ensemble)

$$\langle \mathcal{O}[f(x,Q^2)] \rangle = \frac{1}{N_{\rm rep}} \sum_{k=1}^{N_{\rm rep}} \mathcal{O}[f^{(k)}(x,Q^2)]$$

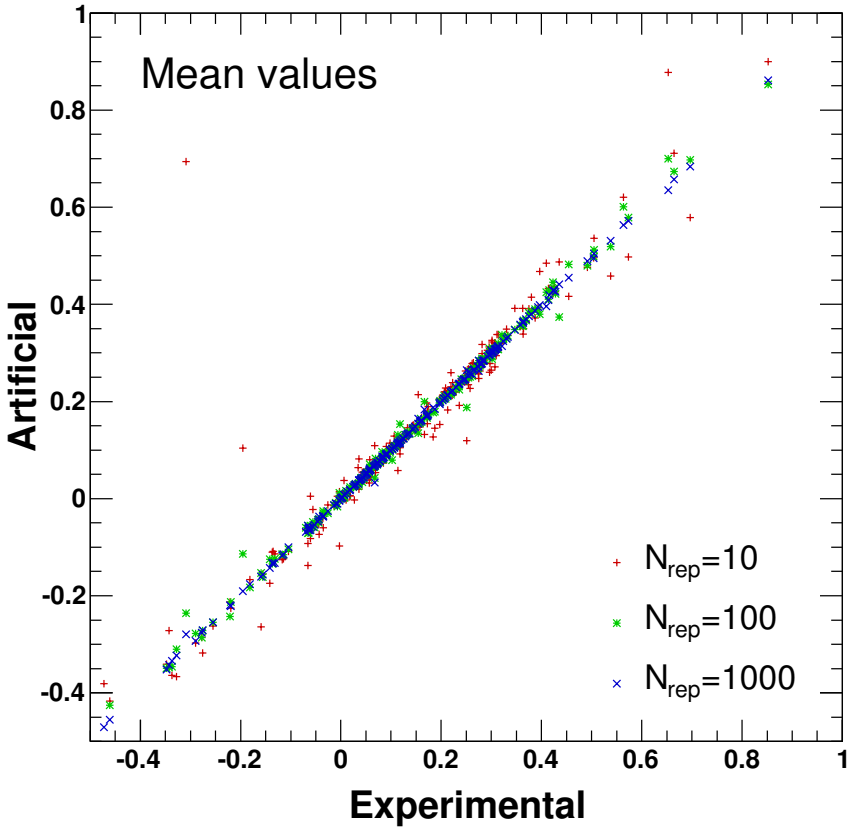
$$\sigma_{\mathcal{O}}[f(x,Q^2)] = \left[\frac{1}{N_{\rm rep}-1} \sum_{k=1}^{N_{\rm rep}} \left(\mathcal{O}[f^{(k)}(x,Q^2)] - \langle \mathcal{O}[f(x,Q^2)] \rangle \right)^2 \right]^{1/2}$$

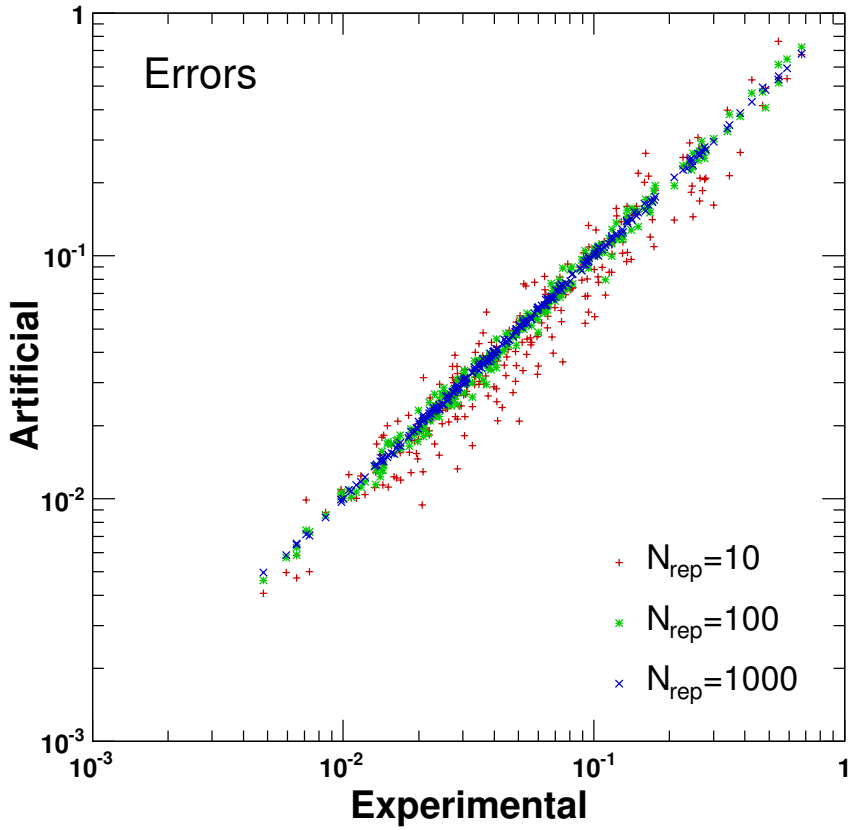
$\Rightarrow$ **no need to rely on linear approximation**

$\Rightarrow$ **computational expensive: need to perform $N_{\rm rep}$ fits instead of one**

The Monte Carlo method: determining the sample size

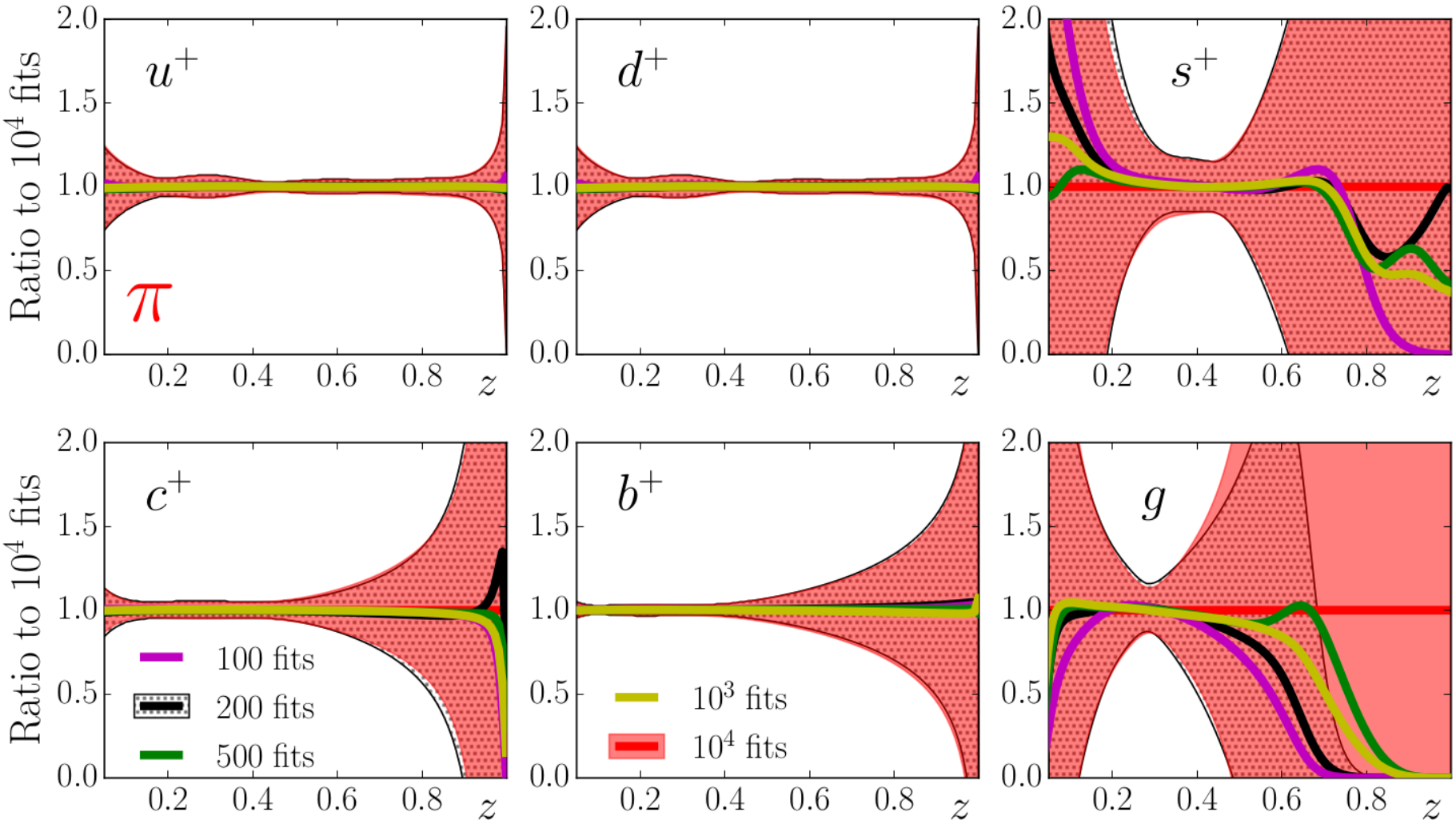
Require that the average over the replicas reproduces the central value of the original experimental data to a desired accuracy (the standard deviation reproduces the error and so on)





Accuracy of few % requires ~ 100 replicas

The Monte Carlo method: determining the sample size



FFs from 200 posterior samples are comparable with those with 10^4 samples

The Monte Carlo method: importance sampling

1. Generalise the way in which artificial replicas are generated

$$\mathcal{O}_i^{(\mathrm{art}),(k)}(x,Q^2) = \left[1 + \sum_c r_{c,i}^{(k)} \sigma_{c,i} + r_{s,i}^{(k)} \sigma_{s,i}\right] \mathcal{O}_i^{(\mathrm{exp})}(x,Q^2)$$

$\sigma_{c,p}$: correlated systematics

$\sigma_{s,p}$: statistical errors (also uncorrelated systematics)

$r_{c,p}^{(k)}$, $r_{s,p}^{(k)}$: Gaussian random numbers

2. Define proper estimators to determine the sample size

	$\left\langle \mathrm{PE}\left[\langle\mathcal{O}^{(\mathrm{art})}\rangle\right]\right\rangle$ [%]			$r\left[\mathcal{O}^{(\mathrm{art})}\right]$		
N_{rep}	10	100	1000	10	100	1000
Exp.1	23.7	3.5	2.9	.76037	.99547	.99712
Exp. 2	19.4	5.6	1.2	.94789	.99908	.99993
...	...	...	...	...	...	...

$$\left\langle PE\left[\langle\mathcal{O}^{(\mathrm{art})}\rangle_{\mathrm{rep}}\right]\right\rangle_{\mathrm{dat}} = \frac{1}{N_{\mathrm{dat}}} \sum_{i=1}^{N_{\mathrm{dat}}} \left| \frac{\langle\mathcal{O}_{\mathrm{i}}^{(\mathrm{art})}\rangle_{\mathrm{rep}} - \mathcal{O}_{\mathrm{i}}^{(\mathrm{exp})}}{\mathcal{O}_{\mathrm{i}}^{(\mathrm{exp})}} \right|$$

Percentage Error

$$r\left[\mathcal{O}^{(\mathrm{art})}\right] = \frac{\left\langle \mathcal{O}^{(\mathrm{exp})} \langle\mathcal{O}^{(\mathrm{art})}\rangle_{\mathrm{rep}} \right\rangle_{\mathrm{dat}} - \langle\mathcal{O}^{(\mathrm{exp})}\rangle_{\mathrm{dat}} \left\langle \langle\mathcal{O}^{(\mathrm{art})}\rangle_{\mathrm{rep}} \right\rangle_{\mathrm{dat}}}{\sigma_{\mathrm{s}}^{(\mathrm{exp})} \sigma_{\mathrm{s}}^{(\mathrm{art})}}$$

Scatter Correlation

Monte Carlo to Hessian [EPJC 75 (2015) 369]

1. Find a subset of Monte Carlo replicas $\{\eta_\alpha^{(i)}\}_{i=1,\ldots,N_{\rm eig}} \subset \left\{f_\alpha^{(k)}\right\}$ such that

$$f_{H,\alpha}^{(k)} \equiv f_\alpha^{(0)} + \sum_i^{N_{\rm eig}} a_i^{(k)} (\eta_\alpha^{(i)} - f_\alpha^{(0)}) \,, \qquad k = 1, \ldots, N_{\rm rep} \qquad \alpha = 1, \ldots, N_f$$

2. Sample the replicas at a discrete set of points and construct the covariance matrix

$$({\rm cov})_{ij,\alpha\beta}^f \equiv \frac{N_{\rm rep}}{N_{\rm rep}-1} \left(\left\langle f_\alpha^{(k)}(x_i) \cdot f_\beta^{(k)}(x_j) \right\rangle_{\rm rep} - \left\langle f_\alpha^{(k)}(x_i) \right\rangle_{\rm rep} \left\langle f_\beta^{(k)}(x_i) \right\rangle_{\rm rep} \right)$$

3. Determine the set of coefficients $\{a^{(k)}\}$ by singular value decomposition of

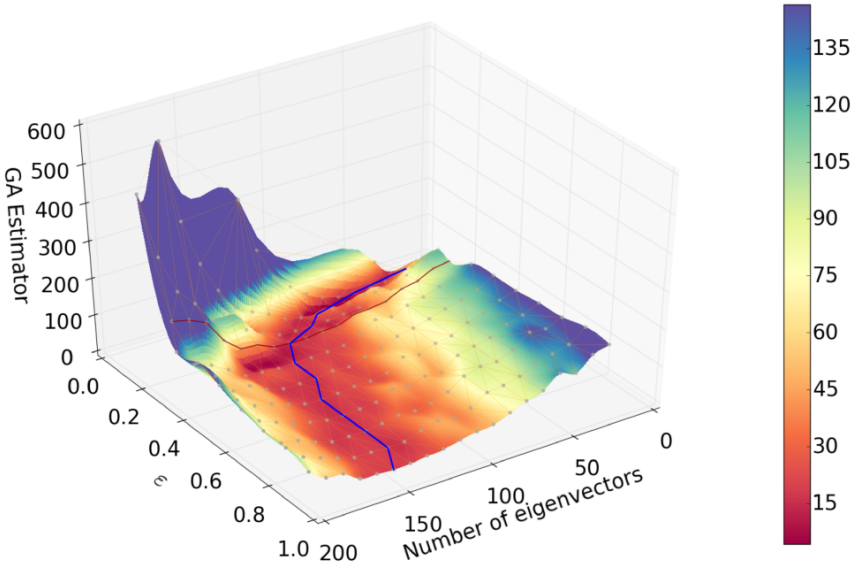
$$\chi_f^{2(k)} = \sum_{i,j}^{N_x} \sum_{\alpha,\beta}^{N_f} \left\{ \left[f_{H,\alpha}^{(k)}(x_i) - f_\alpha^{(k)}(x_i) \right] \left({\rm cov}^f \right)_{ij,\alpha\beta}^{-1} \left[f_{H,\beta}^{(k)}(x_j) - f_\beta^{(k)}(x_j) \right] \right\}$$

4. Determine the Hessian matrix by inverting the covariance matrix

$${\rm cov}_{ij}^{(a)} = \frac{N_{\rm rep}}{N_{\rm rep}-1} \left(\left\langle a_i^{(k)} a_j^{(k)} \right\rangle_{\rm rep} - \left\langle a_i^{(k)} \right\rangle_{\rm rep} \left\langle a_j^{(k)} \right\rangle_{\rm rep} \right)$$

5. Optimise the number of grid points N_x and of eigenvectors $N_{\rm eig}$

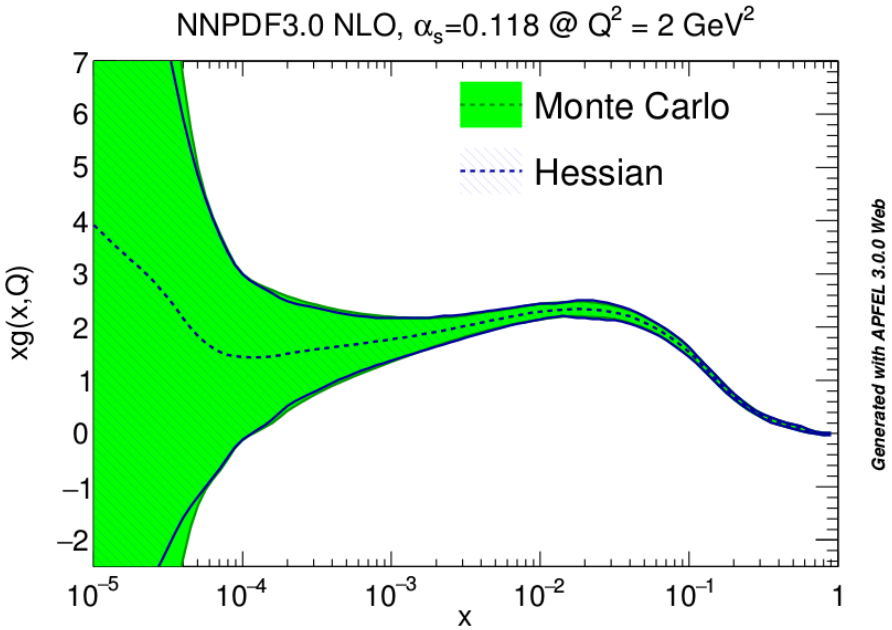
Monte Carlo to Hessian: optimisation and validation

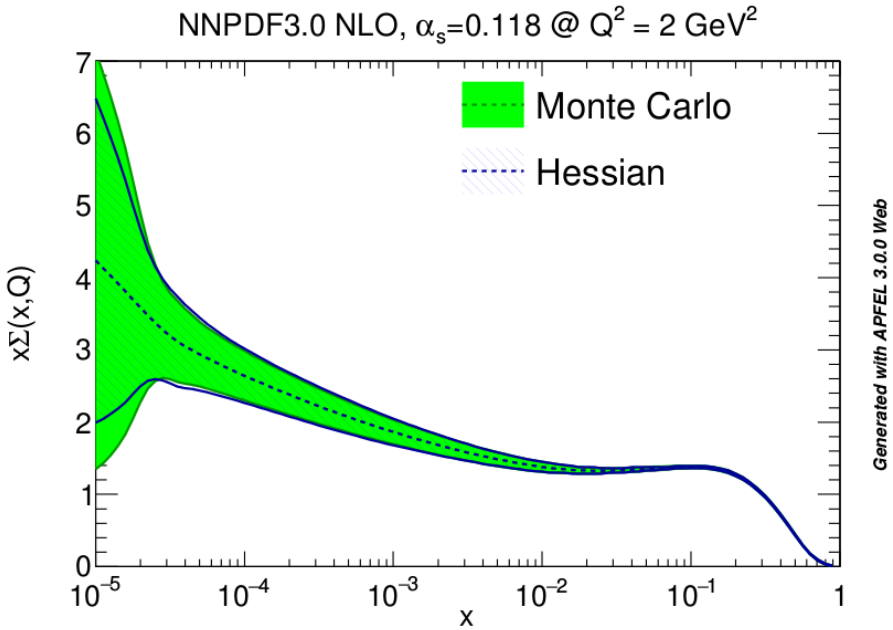


$$\epsilon_\alpha(x_i) = \frac{|\sigma_\alpha(x_i) - \sigma_\alpha^{68}(x_i)|}{\sigma_\alpha^{68}(x_i)} < \epsilon = 0.25$$

$$\text{ERF}_\sigma = \sum_i^{N_x} \sum_\alpha^{N_f} \left| \frac{\sigma_{H,\alpha}^f(x_i) - \sigma_\alpha^f(x_i)}{\sigma_\alpha^f(x_i)} \right|$$

$$N_{\text{rep}} = 1000 \qquad N_{\text{eig}} = 120$$



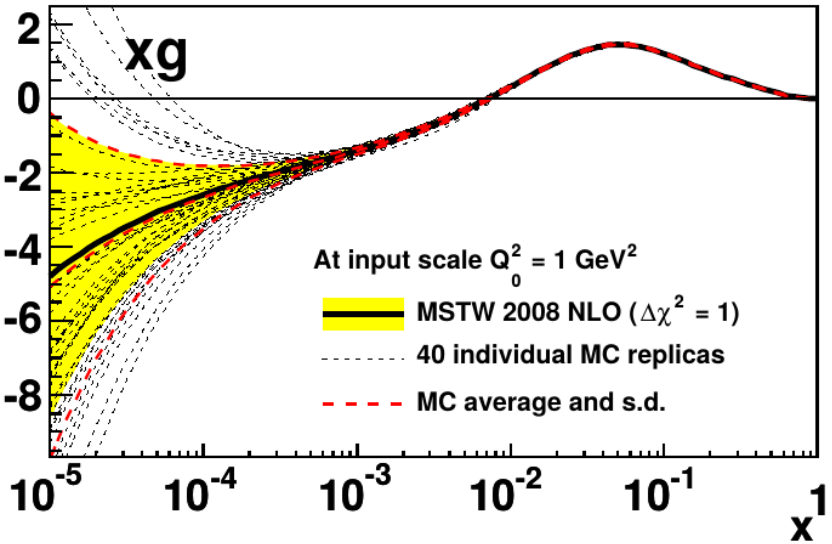


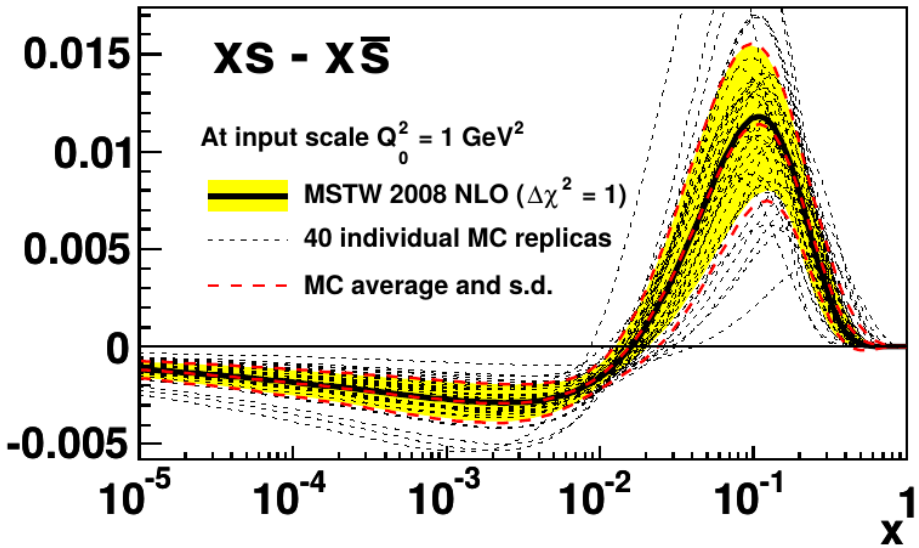
Hessian to Monte Carlo [JHEP 1208 (2012) 052]

1. Generate multi-Gaussian replicas by shifting the data for each data set m

$$D_{i,m} \to \left(D_{i,m} + r_{m,i}^{\text{uncorr.}} \sigma_{m,i}^{\text{uncorr}} + \sum_{k}^{N_{\text{corr}}} r_{m,i,k}^{\text{corr}} \sigma_{m,i,k}^{\text{corr}} \right) \left(1 + r_{m}^{\mathcal{L}} \sigma_{m}^{\mathcal{L}} \right)$$

2. Perform $N_{\text{rep}} = N_{\text{eig}}$ (=40) fits; determine $N_{\text{eig}}/2$ independent parameters

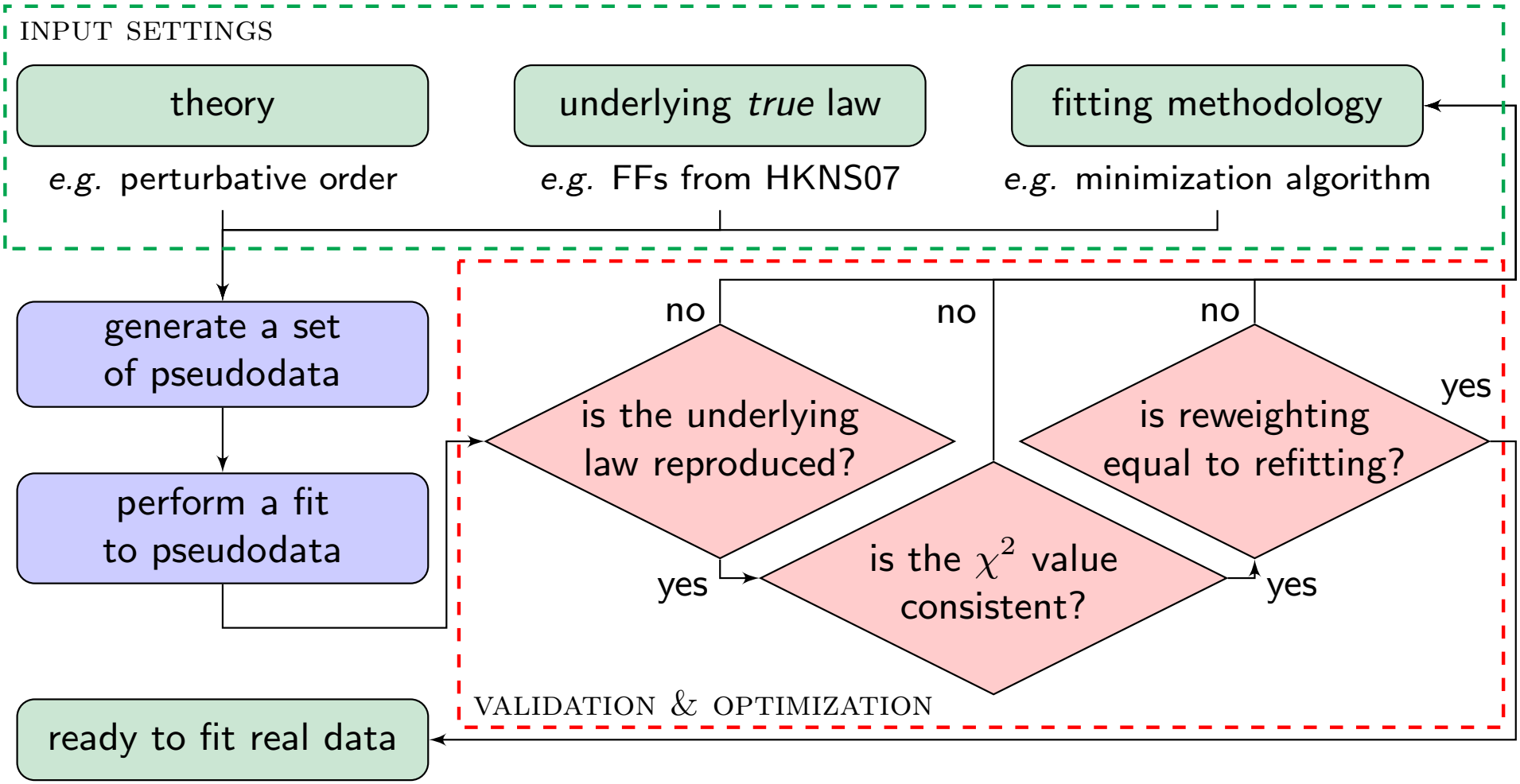




2.6 Closure tests

Closure tests: general idea [JHEP 1504 (2015) 040]

Validation and optimization of the fitting strategy with known underlying physical law



Full control of procedural uncertainties

Closure tests: levels

1. Level 0: generate pseudodata D_i^0 with zero uncertainty
   (but $(\mathrm{cov})_{ij}$ in the χ^2 is the data covariance matrix)
   $\rightarrow$ fit quality can be arbitrarily good, if the fitting methodology is efficient: $\chi^2/N_{\mathrm{dat}} \sim 0$
   $\rightarrow$ validate fitting methodology (parametrisation, minimisation)
   $\rightarrow$ interpolation and extrapolation uncertainty
2. Level 1: generate pseudodata D_i^1 with stochastic fluctuations (no replicas)

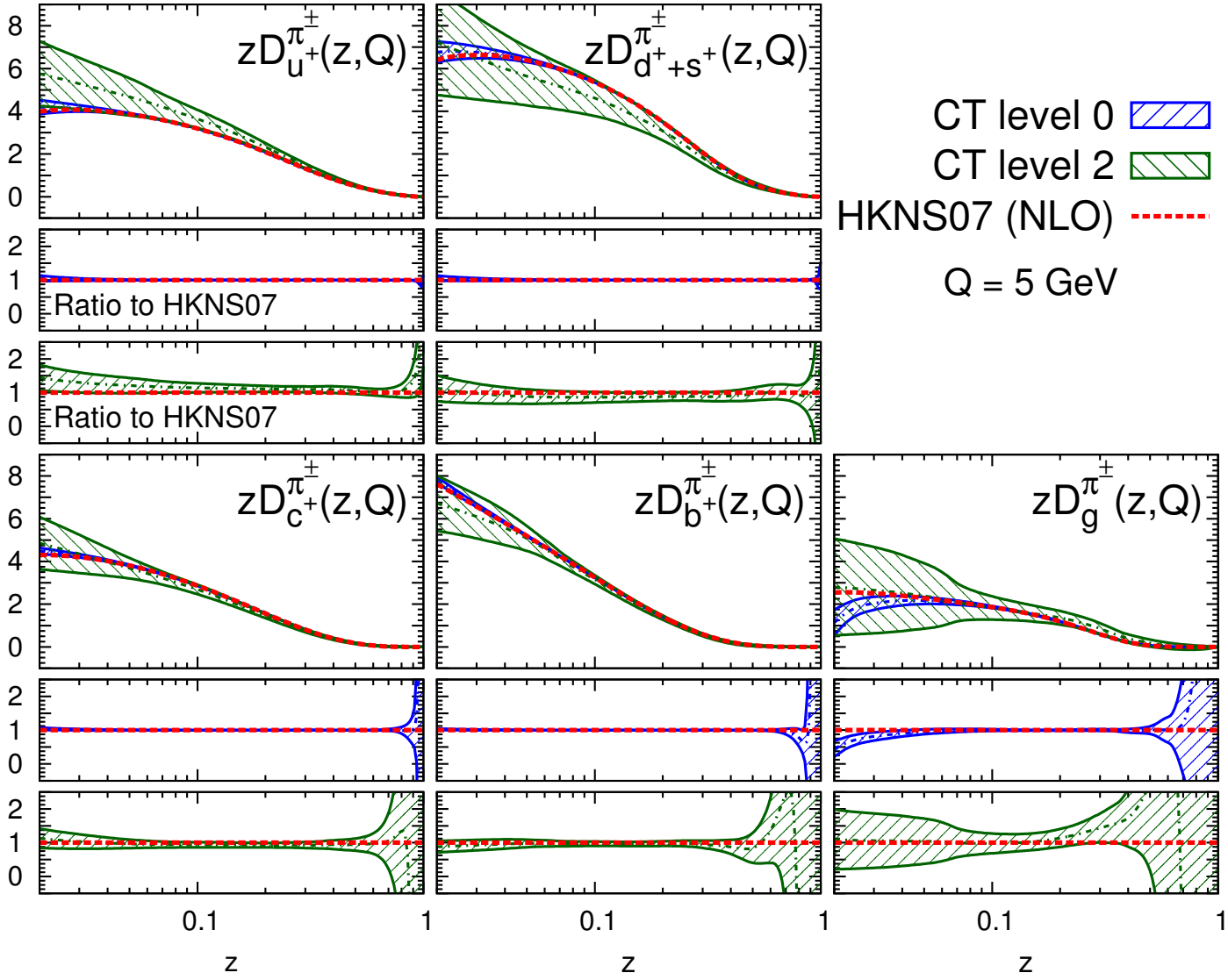
$$D_i^1 = (1 + r_i^{\mathrm{nor}} \sigma_i^{\mathrm{nor}}) \left(D_i^0 + \sum_p^{N_{\mathrm{sys}}} r_{i,p}^{\mathrm{sys}} \sigma_{i,p}^{\mathrm{sys}} + r_i^{\mathrm{stat}} \sigma_i^{\mathrm{stat}} \right)$$

   $\rightarrow$ experimental uncertainties are not propagated into FFs: $\chi^2/N_{\mathrm{dat}} \sim 1$
   $\rightarrow$ functional uncertainty (a large number of functional forms with equally good χ^2)
3. Level 2: generate N_{rep} Monte Carlo pseudodata replicas $D_i^{2,k}$ on top of Level 2

$$D_i^{2,k} = (1 + r_i^{\mathrm{nor},k} \sigma_i^{\mathrm{nor}}) \left(D_i^1 + \sum_p^{N_{\mathrm{sys}}} r_{i,p}^{\mathrm{sys},k} \sigma_{i,p}^{\mathrm{sys}} + r_i^{\mathrm{stat},k} \sigma_i^{\mathrm{stat}} \right)$$

   $\rightarrow$ propagate the fluctuations due to experimental uncertainties into FFs: $\chi^2/N_{\mathrm{dat}} \sim 1$
   $\rightarrow$ input FFs lie within the one-sigma band of the fitted FFs with a probability of $\sim 68\%$
   $\rightarrow$ data uncertainty (tolerance supplement data unc. with extrapolation/functional uncs.)

Closure tests: example



$\chi^2/N_{\rm dat} = 0.0001$ (L0) $\qquad$ $\chi^2/N_{\rm dat} = 1.0262$ (L1)

2.7 Reweighting and unweighting

Reweighting [PRD 58 (1998) 094023]

Assess the impact of including a new data set $\{y\} = \{y_1, \ldots, y_n\}$ in an old PDF set

Bayesian reweighting [NPB 849 (2011) 112, NPB 855 (2012) 608]

1. Evaluate the agreement between new data and each replica f_k in a prior ensemble

$$\chi^2_k(\{y\},\{f_k\}) = \sum_{i,j}^{n} \{y_i - y_i[f_k]\}\, \sigma_{ij}\, \{y_j - y_j[f_k]\}$$

2. Apply Bayes theorem to determine the conditional probability of PDF upon the inclusion of the new data and update the probability density in the space of PDFs

$$\mathcal{P}_{\rm new} = \mathcal{N}_\chi \mathcal{P}(\chi^2_k|\{f_k\})\mathcal{P}_{\rm old}(\{f_k\}) \quad \mathcal{P}(\chi^2_k|\{f_k\}) = [\chi^2_k(\{y\},\{f_k\}]^{\frac{1}{2}(n-1)} e^{-\frac{1}{2}\chi^2_k(\{y\},\{f_k\})}$$

3. Replicas are no longer equally probable. Expectation values are given by

$$\langle \mathcal{O}[f_i(x,Q^2]\rangle_{\rm new} = \sum_{k=1}^{N_{\rm rep}} w_k \mathcal{O}[f_i^{(k)}(x,Q^2)]$$

$$w_k \propto [\chi^2_k(\{y\},\{f_k\})]^{\frac{1}{2}(n-1)} e^{-\frac{1}{2}\chi^2_k(\{y\},\{f_k\})} \quad \text{with} \quad N_{\rm rep} = \sum_{k=1}^{N_{\rm rep}} w_k$$

Reweighting [PRD 58 (1998) 094023]

Assess the impact of including a new data set $\{y\} = \{y_1, \ldots, y_n\}$ in an old PDF set

Bayesian reweighting [NPB 849 (2011) 112, NPB 855 (2012) 608] (Hessian RW see [JHEP 1412 (2014) 100])

1. Evaluate the agreement between new data and each replica f_k in a prior ensemble

$$\chi^2_k(\{y\},\{f_k\}) = \sum_{i,j}^{n} \{y_i - y_i[f_k]\}\, \sigma_{ij}\, \{y_j - y_j[f_k]\}$$

2. Apply Bayes theorem to determine the conditional probability of PDF upon the inclusion of the new data and update the probability density in the space of PDFs

$$\mathcal{P}_{\rm new} = \mathcal{N}_\chi \mathcal{P}(\chi^2_k|\{f_k\})\mathcal{P}_{\rm old}(\{f_k\}) \quad \mathcal{P}(\chi^2_k|\{f_k\}) = [\chi^2_k(\{y\},\{f_k\}]^{\frac{1}{2}(n-1)} e^{-\frac{1}{2}\chi^2_k(\{y\},\{f_k\})}$$

3. Replicas are no longer equally probable. Expectation values are given by

$$\langle \mathcal{O}[f_i(x,Q^2]\rangle_{\rm new} = \sum_{k=1}^{N_{\rm rep}} w_k \mathcal{O}[f_i^{(k)}(x,Q^2)]$$

$$w_k \propto [\chi^2_k(\{y\},\{f_k\})]^{\frac{1}{2}(n-1)} e^{-\frac{1}{2}\chi^2_k(\{y\},\{f_k\})} \quad \text{with} \quad N_{\rm rep} = \sum_{k=1}^{N_{\rm rep}} w_k$$

Unweighting [NPB 855 (2012) 608]

Unweighting allows for constructing an ensemble of equally probable PDFs statistically equivalent to a given reweighted set
Hence, the new set can be given without weights

IDEA

Given a weighted set of $N_{\rm rep}$ replicas, select (possibly more than once) replicas carrying relatively hight weight and discard replicas carrying relatively small weight

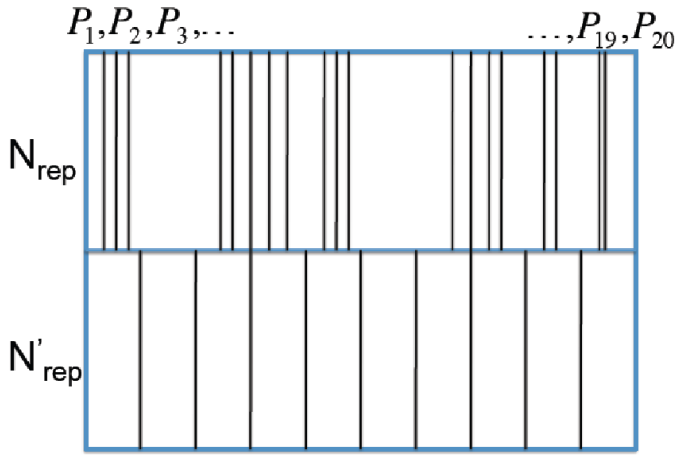
CONSTRUCTION OF THE UNWEIGHTED SET

1. Set the number of replicas $N'_{\rm rep}$ in the unweighted set
   (pointless to choose $N'_{\rm rep} > N_{\rm rep}$: no gain of information)
2. Compute, for the k-th replica of the reweighted set, the integer number

$$w'_k = \sum_{j=1}^{N'_{\rm rep}} \theta\left(\frac{j}{N'_{\rm rep}} - P_{k-1}\right)\theta\left(P_k - \frac{j}{N'_{\rm rep}}\right), \quad P_k = \sum_{j=0}^{k}\frac{w_j}{N_{\rm rep}}, \quad \sum_{k=1}^{N_{\rm rep}} w'_k = N'_{\rm rep}$$

3. Construct the unweighted set taking w'_k copies of the k-th replica, $k = 1, \ldots, N_{\rm rep}$

Unweighting [NPB 855 (2012) 608]



CONSTRUCTION OF THE UNWEIGHTED SET

1. Set the number of replicas $N'_{\rm rep}$ in the unweighted set
   (pointless to choose $N'_{\rm rep} > N_{\rm rep}$: no gain of information)
2. Compute, for the k-th replica of the reweighted set, the integer number

$$w'_k = \sum_{j=1}^{N'_{\rm rep}} \theta\left(\frac{j}{N'_{\rm rep}} - P_{k-1}\right) \theta\left(P_k - \frac{j}{N'_{\rm rep}}\right), \quad P_k = \sum_{j=0}^{k} \frac{w_j}{N_{\rm rep}}, \quad \sum_{k=1}^{N_{\rm rep}} w'_k = N'_{\rm rep}$$

3. Construct the unweighted set taking w'_k copies of the k-th replica, $k = 1, \ldots, N_{\rm rep}$

2.8 Summary of Lecture 2

Summary

1. Determine the probability density in the space of FFs
   $\longrightarrow$ the problem is projected onto the finite-dimensional space of parameters
   $\longrightarrow$ requires a parametrisation: standard vs flexible (neural networks)
   $\longrightarrow$ requires an algorithm to determine best-fit parameters
2. Represent FF errors
   $\longrightarrow$ carefully handle correlated/uncorrelated and additive/multiplicative uncertainties
   $\longrightarrow$ Hessian vs Monte Carlo
   $\longrightarrow$ Monte Carlo very flexible, Hessian very efficient
   $\longrightarrow$ conversion possible either way
3. Validate fitting methodology
   $\longrightarrow$ performed in the Monte Carlo approach
   $\longrightarrow$ full characterisation of uncertainties
4. Evaluate the impact of data without refitting: reweighting
5. Methodologies used by each active collaboration to determine FFs
   DEHSS, HKKS: standard parametrisation + Hessian
   JAM: standard parametrisation + (iterative) Monte Carlo + cross-validation
   NNPDF: neural network parametrisation + Monte Carlo + GA + cross-validation

Lecture 3: Recent results in the determination of FFs and applications of FFs