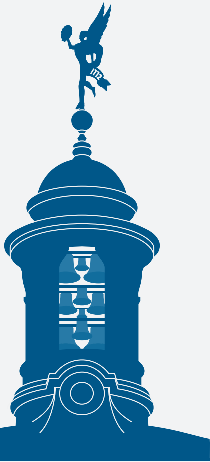
Proton PDF Uncertainties at NNLO from a Markov chain Monte Carlo Investigation

PDFLattice 2024

Peter Risse (prisse@smu.edu)



nCTEQ
nuclear parton distribution functions

based on: 2407:12377 (DIS24 proceedings)

Workshop goals addressed in this talk

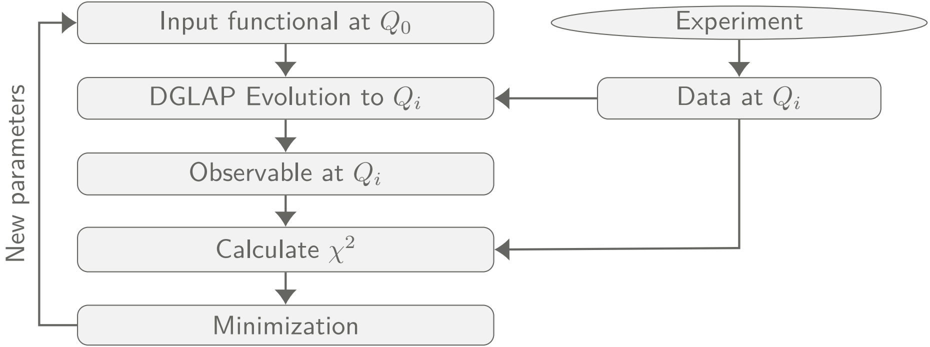
1. Accessing PDFs: lattice and pheno approaches

- **B** How does a **phenomenological fit** (global analysis) assess PDFs using a data-driven methodology grounded in the QCD factorization formalism?
- **C** What are the current efforts, directions, and challenges in both **lattice and pheno/global analyses**? How can we foster **synergy** by establishing a common language between them?

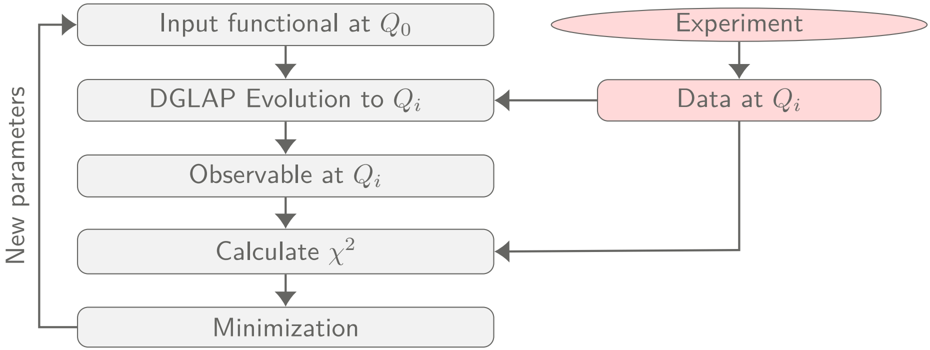
4. Uncertainty Quantification (UQ) and bias/variance tradeoff

- **B** How do we **propagate uncertainties** using methods such as bootstrap, importance sampling, and the Hessian formalism?

Typical minimization procedure

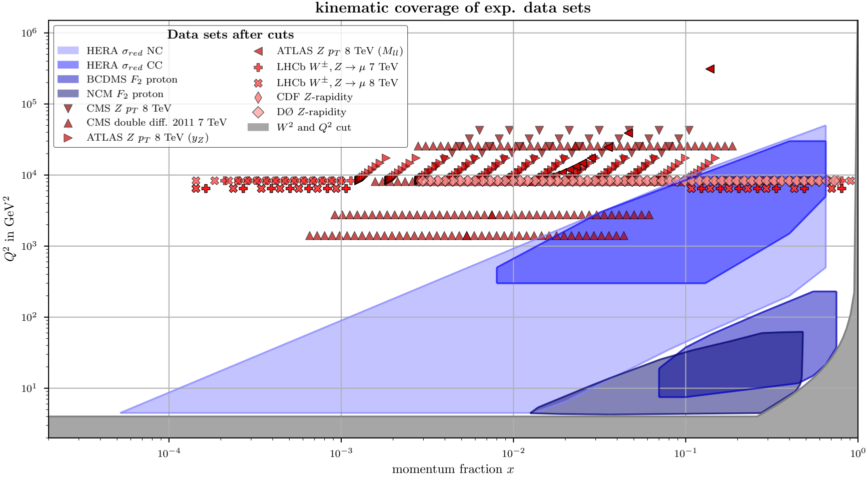


Typical minimization procedure

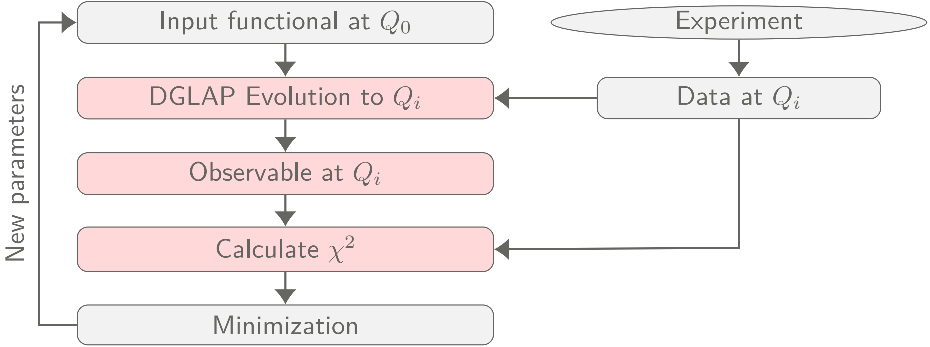


Experimental data

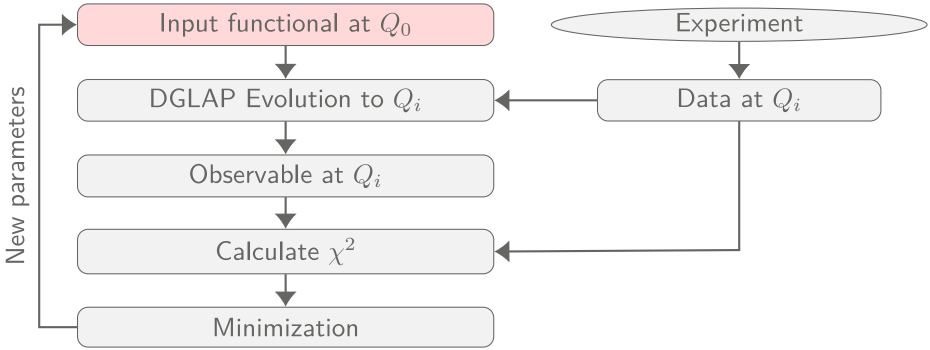
- DIS: 1660 points
  - HERA NC/CC
  - NMC F_2
  - BCDMS F_2
- DY: 324 points
  - CDF & DØ
  - CMS
  - ATLAS
  - LHCb
- Total: 1984 points



Typical minimization procedure



Typical minimization procedure



Input functional form

Functional form

$$f_i(x, Q_0) = \mathbf{c_0} x^{\mathbf{c_1}} (1-x)^{\mathbf{c_2}} (1 + \mathbf{c_3}\sqrt{x} + \mathbf{c_4} x) \qquad Q_0 = 1.3\,\text{GeV}$$

Input functional form

Functional form

$$f_i(x, Q_0) = \mathbf{c_0} x^{\mathbf{c_1}} (1-x)^{\mathbf{c_2}} (1 + \mathbf{c_3}\sqrt{x} + \mathbf{c_4} x) \qquad Q_0 = 1.3\,\text{GeV}$$

Flavor-combinations

$\mathbf{u_v}$	$\rightarrow$	c_1	c_2	c_4		
$\mathbf{d_v}$	$\rightarrow$	c_1	c_2	c_4		
$\bar{\mathbf{u}} + \bar{\mathbf{d}}$	$\rightarrow$	c_1	c_2	c_4		
$\mathbf{s} + \bar{\mathbf{s}}$	$\rightarrow$	c_0				
$\mathbf{g}$	$\rightarrow$	c_0	c_1	c_2	c_3	c_4

Total: 15 parameters

Input functional form

Functional form

$$f_i(x, Q_0) = \mathbf{c_0} x^{\mathbf{c_1}} (1-x)^{\mathbf{c_2}} (1 + \mathbf{c_3}\sqrt{x} + \mathbf{c_4} x) \qquad Q_0 = 1.3\,\text{GeV}$$

Flavor-combinations

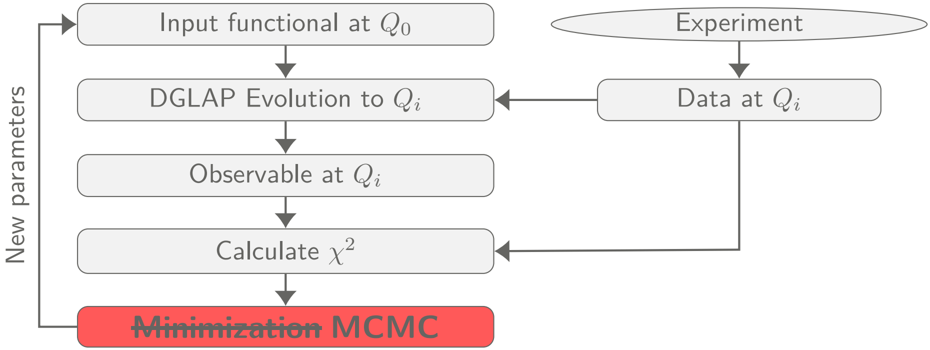
$\mathbf{u_v}$	$\rightarrow$	c_1	c_2	c_4		
$\mathbf{d_v}$	$\rightarrow$	c_1	c_2	c_4		
$\mathbf{\bar{u} + \bar{d}}$	$\rightarrow$	c_1	c_2	c_4		
$\mathbf{s + \bar{s}}$	$\rightarrow$	c_0				
$\mathbf{g}$	$\rightarrow$	c_0	c_1	c_2	c_3	c_4

Total: 15 parameters

Result

$$\chi^2/\text{d.o.f.} = 2380.25/1969 = 1.20$$

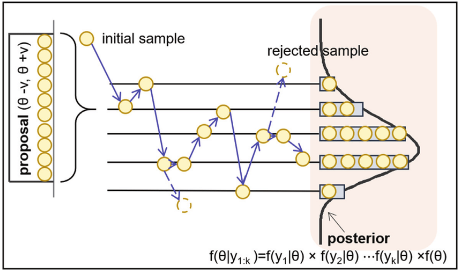
The Markov chain Monte Carlo approach



Markov chain Monte Carlo representation of the posterior

- **draw random samples** from the posterior function

$$\mathrm{post}(\mathbf{c}|D) = \frac{1}{\mathcal{Z}} \exp\left(-\frac{1}{2}\chi^2(\mathbf{c}, D)\right)$$
$$\rightarrow \{\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_n}\}$$



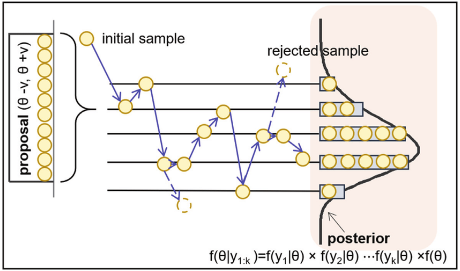
Markov chain Monte Carlo representation of the posterior

- **draw random samples** from the posterior function

$$\mathrm{post}(\mathbf{c}|D) = \frac{1}{\mathcal{Z}} \exp\left(-\frac{1}{2}\chi^2(\mathbf{c}, D)\right)$$
$$\rightarrow \{\mathbf{c_1}, \mathbf{c_2}, \ldots, \mathbf{c_n}\}$$

- samples have to **reproduce the expectation value and higher modes**

$$E\{\mathcal{O}(\mathbf{c})\} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{O}(\mathbf{c}_i)$$



Markov chain Monte Carlo representation of the posterior

- construct the Monte Carlo samples via a Markov chain

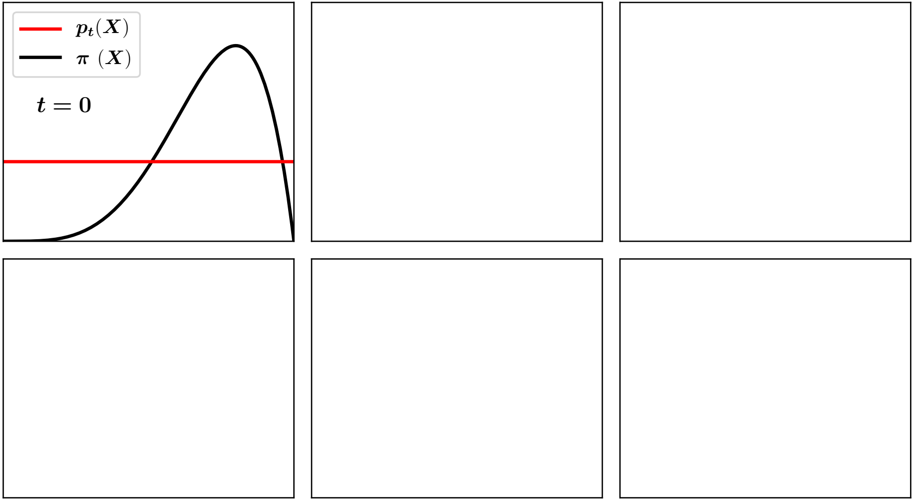
$$\{\mathbf{c}_1 \to \mathbf{c}_2 \to \cdots \to \mathbf{c}_{n-1} \to \mathbf{c}_n\}$$

with $$p_i(\mathbf{c}) = \int \mathrm{d}\mathbf{c}'\, p_{i-1}(\mathbf{c}') T(\mathbf{c}', \mathbf{c})$$

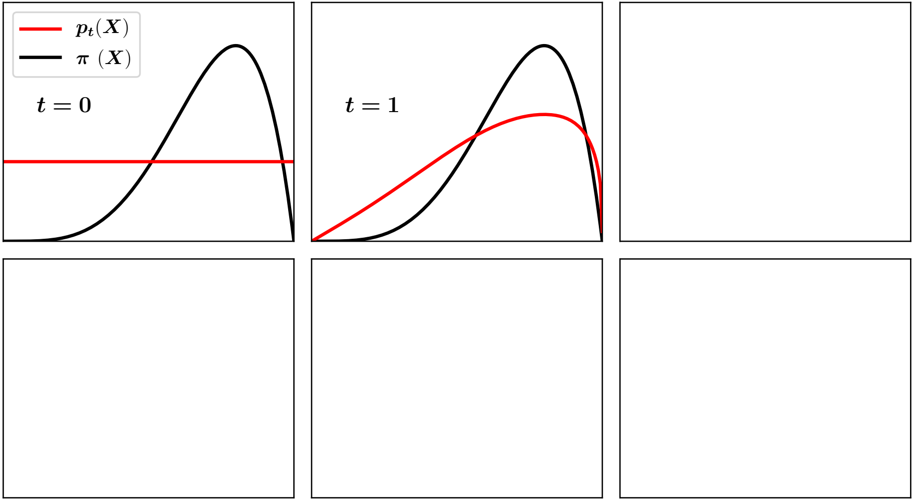
- with the **transition kernel** $T(\mathbf{c}, \mathbf{c}')$

$$\underbrace{p_t(\mathbf{c}) \quad \xrightarrow{t\to\infty} \quad \mathrm{post}(\mathbf{c}|D)}_{\text{proper MCMC algorithm: } T(\mathbf{c},\mathbf{c}')}$$

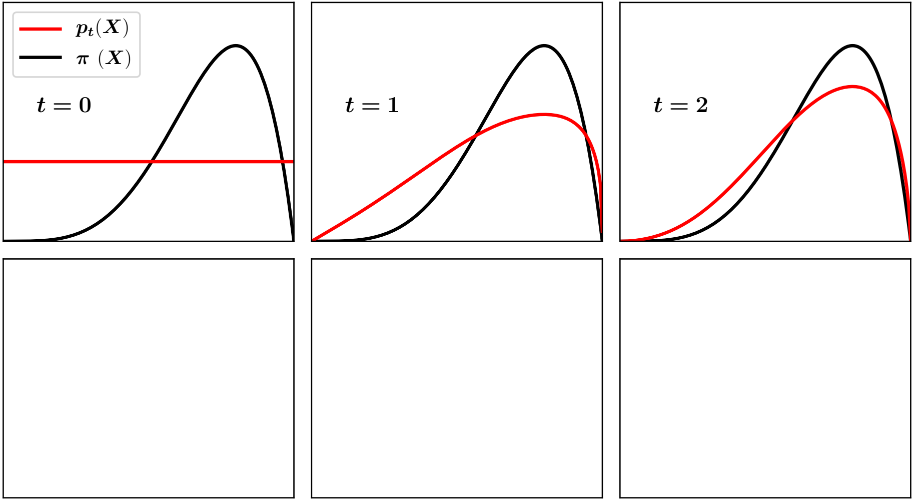
Markov chain Monte Carlo representation of the posterior



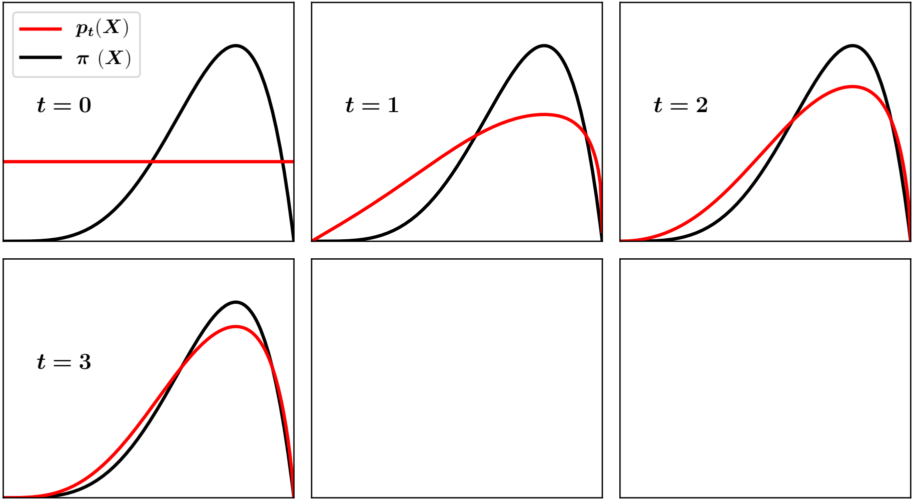
Markov chain Monte Carlo representation of the posterior



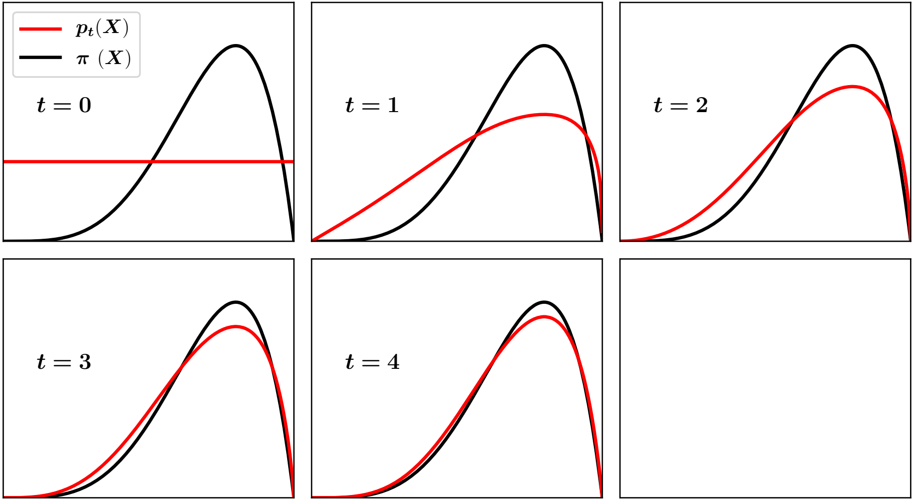
Markov chain Monte Carlo representation of the posterior



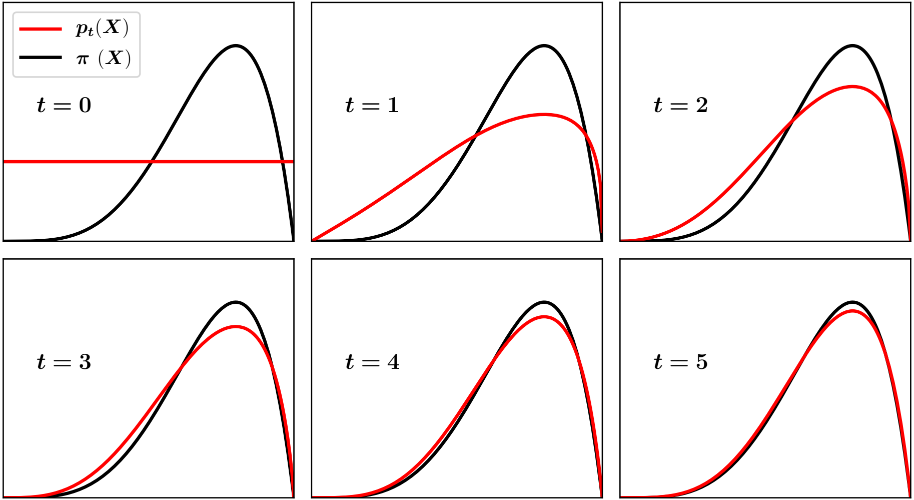
Markov chain Monte Carlo representation of the posterior



Markov chain Monte Carlo representation of the posterior



Markov chain Monte Carlo representation of the posterior



Choosing the proposal distribution – Adaptive Metropolis-Hastings

1. Use **normal random walk Metropolis-Hastings** until N_0 samples have been obtained
   - proposal distribution: multivariate Gaussian

$$\tilde{\mathbf{c}}_{i+1} \quad \text{proposed from} \quad q(\tilde{\mathbf{c}}_{i+1}, \mathbf{c}_i) = \mathcal{N}(\mathbf{c}_i, C_0) \quad \text{with} \quad C_0 : \text{ covariance matrix from user input}$$

H. Haario et al.: "An adaptive Metropolis algorithm", *Bernoulli* 7.2 (Apr. 2001)

Choosing the proposal distribution – Adaptive Metropolis-Hastings

1. Use **normal random walk Metropolis-Hastings** until N_0 samples have been obtained
   - proposal distribution: multivariate Gaussian

$$\tilde{\mathbf{c}}_{i+1} \quad \text{proposed from} \quad q(\tilde{\mathbf{c}}_{i+1}, \mathbf{c}_i) = \mathcal{N}(\mathbf{c}_i, C_0) \quad \text{with} \quad C_0 : \text{ covariance matrix from user input}$$

2. switch to a **self learning proposal distribution**

$$\tilde{\mathbf{c}}_{i+1} \quad \text{proposed from} \quad q(\tilde{\mathbf{c}}_{i+1}, \mathbf{c}_i) = (1-\beta)\mathcal{N}\left(\mathbf{c}_i, \text{scale} \cdot \overline{C}_i\right) + \beta\mathcal{N}(\mathbf{c}_i, C_0)$$

with self learned $\overline{\mathbf{C}}_\mathbf{i}$

- $0 \leq \beta \leq 1$ controls the impact of the 'learned' proposal

H. Haario et al.: "An adaptive Metropolis algorithm", *Bernoulli* 7.2 (Apr. 2001)

Choosing the proposal distribution – Adaptive Metropolis-Hastings

1. Use **normal random walk Metropolis-Hastings** until N_0 samples have been obtained
   - proposal distribution: multivariate Gaussian

$$\tilde{\mathbf{c}}_{i+1} \quad \text{proposed from} \quad q(\tilde{\mathbf{c}}_{i+1}, \mathbf{c}_i) = \mathcal{N}(\mathbf{c}_i, C_0) \quad \text{with} \quad C_0: \text{ covariance matrix from user input}$$

2. switch to a **self learning proposal distribution**

$$\tilde{\mathbf{c}}_{i+1} \quad \text{proposed from} \quad q(\tilde{\mathbf{c}}_{i+1}, \mathbf{c}_i) = (1-\beta)\mathcal{N}\left(\mathbf{c}_i, \text{scale} \cdot \overline{C}_i\right) + \beta\mathcal{N}(\mathbf{c}_i, C_0)$$

with self learned $\overline{\mathbf{C}}_\mathbf{i}$

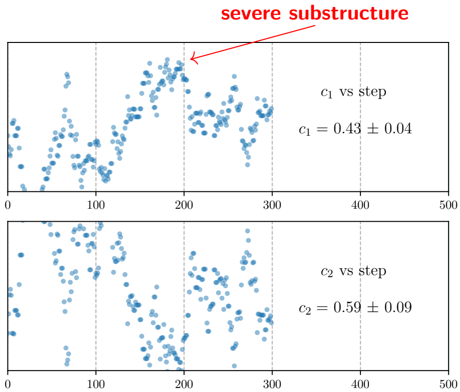
   - $0 \leq \beta \leq 1$ controls the impact of the 'learned' proposal

3. reset self learned proposal distribution **to boost convergence**
   - this reduces the impact of the starting point

H. Haario et al.: "An adaptive Metropolis algorithm", *Bernoulli* 7.2 (Apr. 2001)

Autocorrelation

- we **cannot use the simple equations** to estimate variances and higher modes
  - these severely underestimate the true PDF-Uncertainties

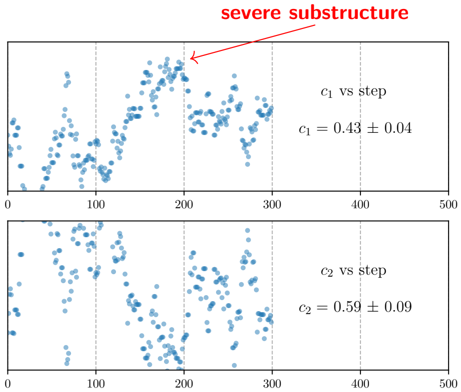


autocorrelation at full force

Autocorrelation

- we **cannot use the simple equations** to estimate variances and higher modes
  - these severely underestimate the true PDF-Uncertainties

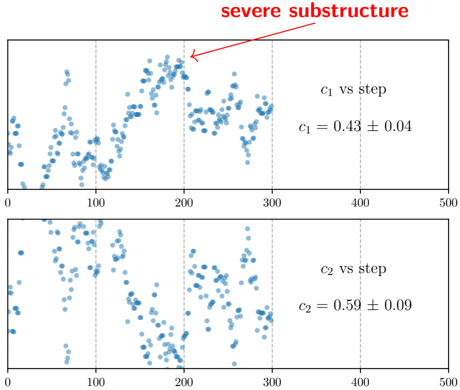
- since every new sample depends on the current **the gain in information is reduced**



autocorrelation at full force

Autocorrelation

- we **cannot use the simple equations** to estimate variances and higher modes
  - these severely underestimate the true PDF-Uncertainties
- since every new sample depends on the current **the gain in information is reduced**
- twice the **autocorrelation-time τ** estimates the number of links in the chain **until the next independent sample is drawn**



autocorrelation at full force

Bridge to Lattice QCD

- lattice QCD uses several methods **dealing with autocorrelation and uncertainty estimation** in general

Bridge to Lattice QCD

- lattice QCD uses several methods **dealing with autocorrelation and uncertainty estimation** in general
- one example is the **Γ-method**
  - this method estimates the autocorrelation time directly from the chain
  - used to **enlarge error estimates** as to eliminate bias
  - **or filter** the time series to get uncorrelated samples

Monte Carlo errors with less errors.

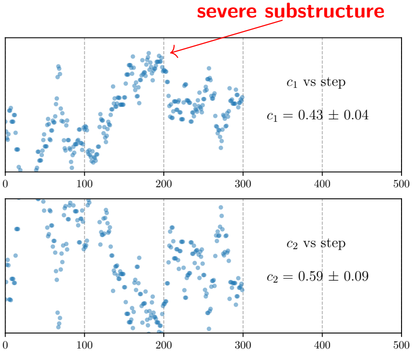
Ulli Wolff*
Institut für Physik, Humboldt Universität
Newtonstr. 15
12489 Berlin, Germany

ALPHA Collaboration

Abstract
We explain in detail how to estimate mean values and assess statistical errors for arbitrary functions of elementary observables in Monte Carlo simulations. The method is to estimate and sum the relevant autocorrelation functions, which is argued to produce more certain error estimates than binning techniques and hence to help toward a

arXiv:hep-lat/0306017

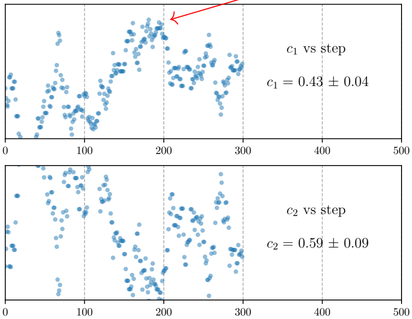
Filtering based on the Γ-method



using 300 samples directly

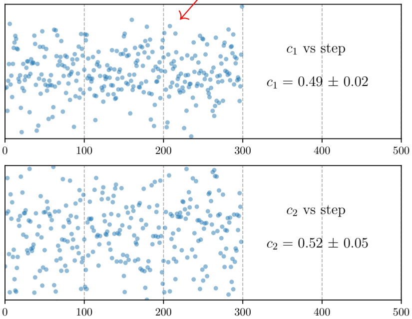
Filtering based on the Γ-method

severe substructure



$\Longrightarrow$

uniform behaviour



using 300 samples directly

thinning 10^4 samples to a total of 300

Markov chain Monte Carlo: Advantages

PDF uncertainty estimation

- statistically sound estimation of uncertainties

Markov chain Monte Carlo: Advantages

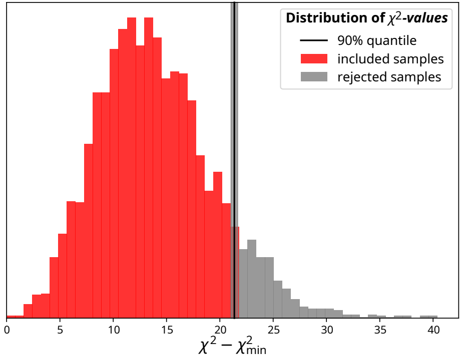
PDF uncertainty estimation

- statistically sound estimation of uncertainties
- directly comparable to Hessian method

Markov chain Monte Carlo: Advantages

PDF uncertainty estimation

- statistically sound estimation of uncertainties
- directly comparable to Hessian method
- estimation of the tolerance T^2



Markov chain Monte Carlo: Advantages and Extensions

PDF uncertainty estimation

- statistically sound estimation of uncertainties
- directly comparable to Hessian method
- estimation of the tolerance T^2

Extensions of current methodology

- improved proposal algorithm
  - Hamilton/Hybrid Monte Carlo (see LQCD!)
- Simulated tempering: addressing the multimodal χ^2-function

Markov chain Monte Carlo: Advantages and Extensions

PDF uncertainty estimation

- statistically sound estimation of uncertainties
- directly comparable to Hessian method
- estimation of the tolerance T^2

Extensions of current methodology

- improved proposal algorithm
  - Hamilton/Hybrid Monte Carlo (see LQCD!)
- Simulated tempering: addressing the multimodal χ^2-function

Thank you for your attention!

backup

A flaw in the Parametrization

Down-valence Distribution

$$xd_v(x,Q_0) = c_0 x^{c_1}(1-x)^{c_2}\left(1 + c_3\sqrt{x} + \mathbf{c_4}x\right)$$

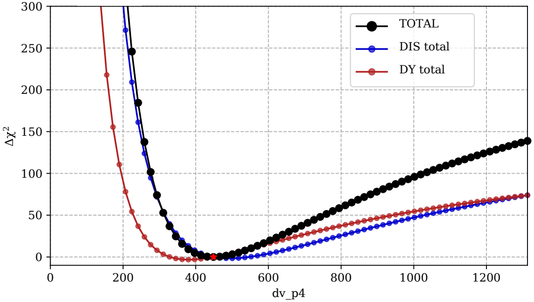
becomes independent of $\mathbf{c_4}$

$$\lim_{\mathbf{c_4}\to\infty} xd_v(x,Q_0) = \lim_{\mathbf{c_4}\to\infty} c_0 x^{c_1}(1-x)^{c_2}\left[\mathbf{c_4}x\right]$$
$$= \tilde{\mathbf{c}}_\mathbf{0} x^{c_1+1}(1-x)^{c_2}$$

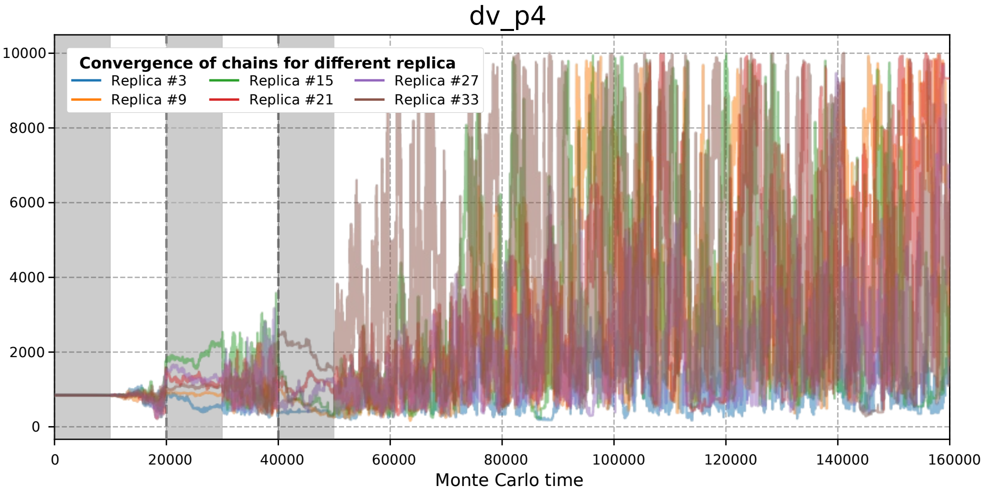
- need constrain $\mathbf{c_4}$ by Uniform Prior:

$$-1000 \leq \mathbf{c_4} \leq 10.000$$

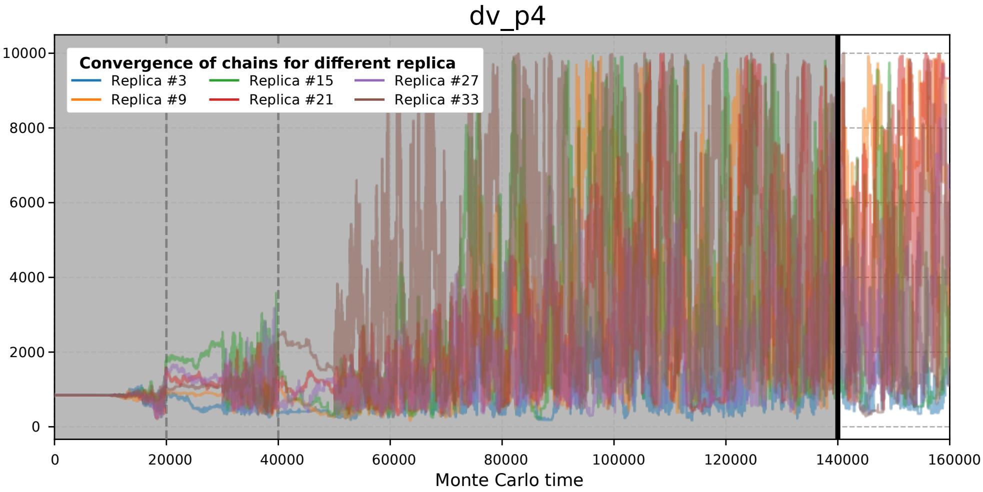
1D parameter scan



Thermalization



Thermalization



Fitting setup

PDF parameters

$$f_i(x,Q_0) = \mathbf{c_0} x^{\mathbf{c_1}}(1-x)^{\mathbf{c_2}}(1+\mathbf{c_3}\sqrt{x}+\mathbf{c_4}x)$$

$\mathbf{u_v}$	$\rightarrow$	c_1	c_2	c_4			
$\mathbf{d_v}$	$\rightarrow$	c_1	c_2	c_4	(Prior)		
$\mathbf{\bar{u}+\bar{d}}$	$\rightarrow$	c_1	c_2	c_4			
$\mathbf{s+\bar{s}}$	$\rightarrow$	c_0					
$\mathbf{g}$	$\rightarrow$	c_0	c_1	c_2	c_3	c_4	

Total: 15 parameters

Hyperparameters

- Proposals: Adaptive Metropolis Hastings
- 36 independent chains with 479.000 samples each
  - burn-in phase: 140.000 samples
  - **Total:** 17 million samples
- removing autocorrelation and burn-in:

Total: 4068 uncorrelated samples

$$\chi^2/\mathbf{d.o.f.} = 2380.25/1969 = 1.20$$

From Samples to PDF-Uncertainties

Confidence interval for observable $\mathcal{O}(c)$

$$\mathcal{O}_- \leq \mathcal{O} \leq \mathcal{O}_+$$

From Samples to PDF-Uncertainties

Confidence interval for observable $\mathcal{O}(c)$

$$\mathcal{O}_- \leq \mathcal{O} \leq \mathcal{O}_+$$

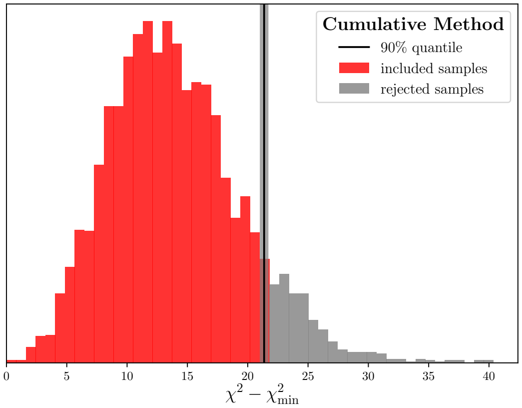
Cumulative χ^2-Method

Central: sample with minimal $\chi^2 \to \mathcal{O}_{\chi^2_{min}}$

Lower bound: $\min(\{\mathcal{O}\}_{90\%})$

Upper bound: $\max(\{\mathcal{O}\}_{90\%})$

A. Putze et al., arXiv: 0808.2437



From Samples to PDF-Uncertainties

Confidence interval for observable $\mathcal{O}(c)$

$$\mathcal{O}_- \leq \mathcal{O} \leq \mathcal{O}_+$$

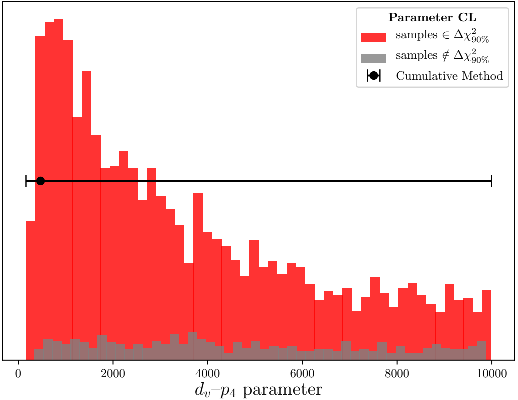
Cumulative χ^2-Method

Central: sample with minimal $\chi^2 \to \mathcal{O}_{\chi^2_{min}}$

Lower bound: $\min(\{\mathcal{O}\}_{90\%})$

Upper bound: $\max(\{\mathcal{O}\}_{90\%})$

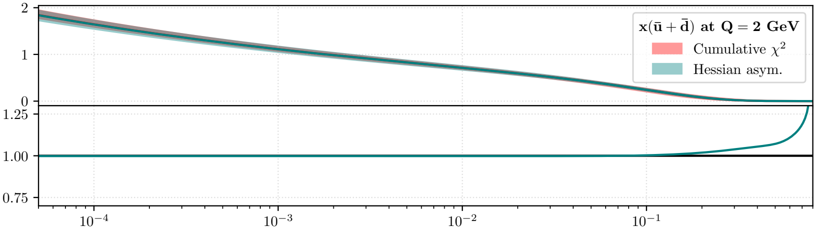
A. Putze et al., arXiv: 0808.2437



Comparison with Hessian – Gaussian parameters

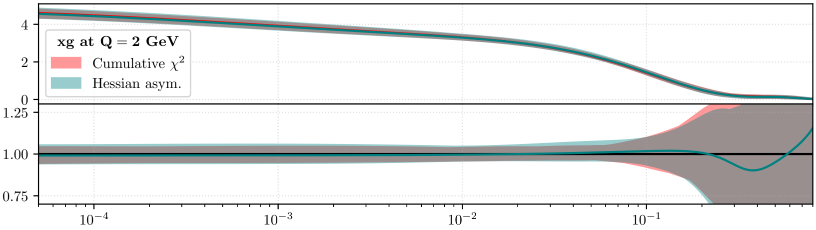
Cumulative χ^2

$$\Delta\chi^2_{90\%} = 22$$



Hessian Method

$$\Delta\chi^2 = 22$$



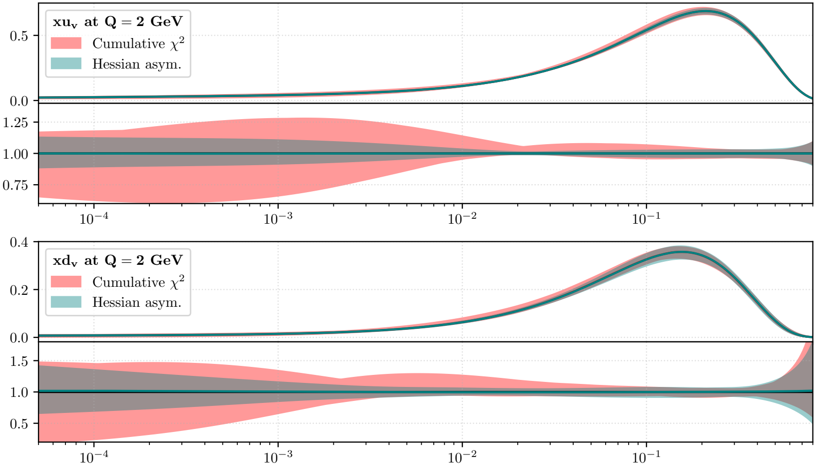
Comparison with Hessian – non-Gaussian parameters

Cumulative χ^2

$$\Delta\chi^2_{90\%} = 22$$

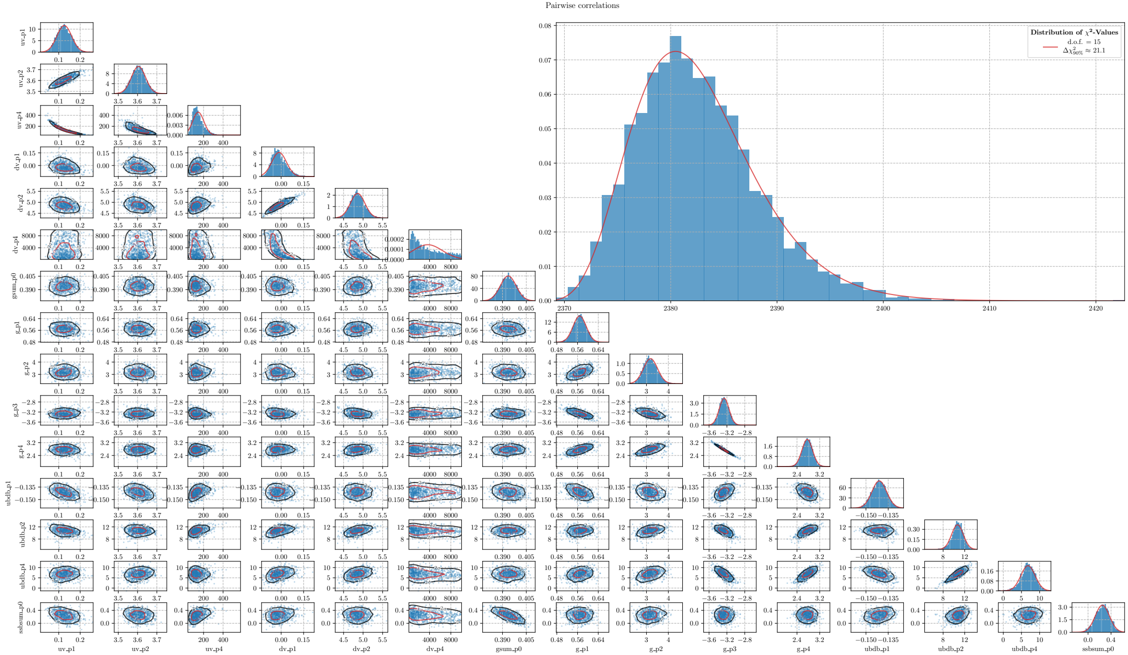
Hessian Method

$$\Delta\chi^2 = 22$$



Description of Experimental Data

Data Set	Ref.	Data Points	χ^2/data
DIS			
HERA σ_{red} neutral current	[54]	1039	1.26
HERA σ_{red} charged current	[54]	81	1.08
BCDMS F_2 proton	[135]	339	1.09
NCM F_2 proton	[136]	201	1.54
DIS total		1660	1.25
DY			
CDF Z-rapidity	[137]	28	1.10
DØ Z-rapidity	[138]	28	0.60
ATLAS Z p_T 8 TeV (M_{ll})	[139]	44	1.06
ATLAS Z p_T 8 TeV (y_Z)	[139]	48	0.65
CMS Z p_T 8 TeV	[140]	28	0.46
CMS double diff. 2011 7 TeV	[141]	88	1.02
LHCb $W^\pm, Z \to \mu$ 7 TeV	[142]	29	1.07
LHCb $W^\pm, Z \to \mu$ 8 TeV	[143]	31	1.18
DY total		324	0.91
Total		1984	**1.20** (per dof)



Pairwise correlations
Distribution of χ^2-Values
d.o.f. = 15
$\Delta\chi^2_{90\%} \approx 21.1$
uv_p1
uv_p2
uv_p4
dv_p1
dv_p2
dv_p4
gsum_p0
g_p1
g_p2
g_p3
g_p4
ubdb_p1
ubdb_p2
ubdb_p4
ssbsum_p0

APFEL++ – A PDF evolution library in c++

Bertone, arXiv:1708.00911

- main author: **V. Bertone**
- rewrite of the Fortran APFEL code
  - used by the NNPDF collaboration

APFEL++ – A PDF evolution library in c++

Bertone, arXiv:1708.00911

- main author: **V. Bertone**
- rewrite of the Fortran APFEL code
  - used by the NNPDF collaboration

Precompute observables

$$F_\lambda(x, Q^2) = \sum_k \int_\chi^1 \frac{\mathrm{d}\xi}{\xi} C_k^\lambda\left(\frac{\chi}{\xi}, \frac{Q}{\mu}, \frac{M_i}{\mu}, \alpha_s(\mu)\right) f_k(\xi, \mu)$$

APFEL++ – A PDF evolution library in c++

Bertone, arXiv:1708.00911

- main author: **V. Bertone**
- rewrite of the Fortran APFEL code
  - used by the NNPDF collaboration

Precompute observables

$$F_\lambda(x, Q^2) = \sum_k \int_\chi^1 \frac{\mathrm{d}\xi}{\xi} C_k^\lambda \left(\frac{\chi}{\xi}, \frac{Q}{\mu}, \frac{M_i}{\mu}, \alpha_s(\mu) \right) f_k(\xi, \mu)$$

Replace with interpolating functions: $\Uparrow$

$$\sum_\alpha^{N_\xi} w_\alpha(\xi) f_k(\xi_\alpha, \mu)$$

APFEL++ – A PDF evolution library in c++

Bertone, arXiv:1708.00911

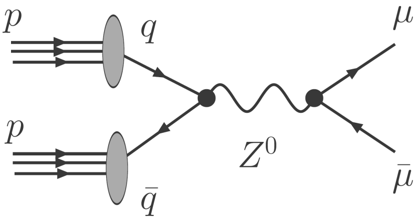
- main author: **V. Bertone**
- rewrite of the Fortran APFEL code
  - used by the NNPDF collaboration

Precompute observables

$$F_\lambda(x, Q^2) = \sum_k \sum_\alpha \underbrace{\int_\chi^1 \frac{\mathrm{d}\xi}{\xi} C_k^\lambda\left(\frac{\chi}{\xi}, \frac{Q}{mu}, \frac{M_i}{\mu}, \alpha_s(\mu)\right) w_\alpha(\xi)}_{\textbf{Precompute}} f_k(\xi_\alpha, \mu)$$

Speed-up of theoretical predictions – Hadron collider

$$\sigma_{pp\to X} = \sum_s^{partons} \sum_p \int \mathrm{d}x_1 \mathrm{d}x_2\, \hat{\sigma}^{(s)(p)} \alpha_s^p(Q^2) F^{(s)}(x_1, x_2, Q^2)\ ,\ F^{(s)} = \sum_{ij} f_i(x_1, Q^2) f_j(x_2, Q^2)$$



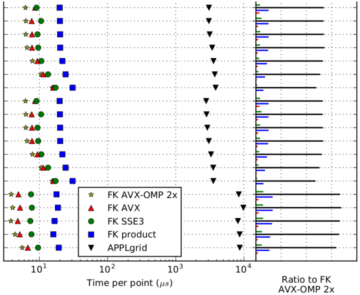
- computationally **expensive double integrals**
- increasing amount of experimental observables
- solution `APPLgrid`
  - interpolate the PDFs
  - precompute the integrals by including the interpolating functions as grids
  - now convolute grids with any pdf to get prediction

T. Carli, D. Clements et al., arXiv:0911.2985

Speed-up of theoretical predictions – Hadron collider

- `APPLgrid` is still too slow for several reasons
  - convolution of the grid with the PDFs is **not well optimized**
  - before one can convolute one has to compute the DGLAP evolution to get the PDFs at every Q

- solution **fast convolution tables** (FK-tables) by `APFELgrid`
  - combines `APPLgrid` tables with DGLAP-evolution tables
    - only need the PDFs at Q_0
  - well optimized by making use of vectorisation and multiprocessing
  - **possible speed-up** compared to `APPLgrid`: $\mathcal{O}(2) - \mathcal{O}(10^3)$



V. Bertone et al., arXiv:1605.02070