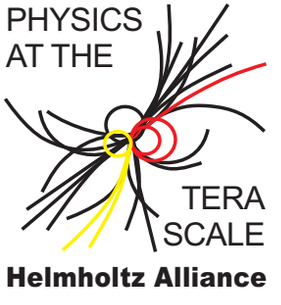
Parameter fitting and PDF4MC

Hendrik Hoeth (Lund)
Albert Knutsson (DESY)
Krzysztof Kutak (DESY)



Overview

Motivation	–	“why” and “what’s the problem”
Strategy	–	“how to tune”
Implementation	–	“which programs are around”
Examples	–	“reality check”
Summary	–	“what we learned”

Motivation – MC tuning

All generators are based on phenomenological models: dipole cascade, string fragmentation, cluster hadronisation, ...

The models have free parameters which are a priori unknown: q_0^2, σ_q, Lund A and B, flavour ratios, ...

We want the MC to describe the data the best possible way. So the parameters need to be tuned.

Even parameters like α_s need to be optimised!

Motivation – PDF fitting

LO and NLO PDFs are using fixed order ME calculations and introduce several problems to MC production:

LO PDFs underestimate jet rates.

NLO PDFs are not physical observables and not really consistent with LO generators.

To get the best possible description of our data, we need to (re)tune the (u)PDF parametrization – in combination with the MC generator.

Problems

The parameters are highly correlated
$\Rightarrow$ can't be tuned one after the other.

Many parameters to be tuned $(\mathcal{O}(10))$.

Tuning all parameters at the same time puts us into a high dimensional parameter space.

Brute force approaches don't work: Running the MC generator takes too long for every point in the parameter space
(= setting of parameters).

We haven't the money, so we've got to think.

– Lord Rutherford

Divide and conquer:

Split the task into parts (parton shower, hadronisation, PDFs) $\Rightarrow$ cut down the number of parameters.

Be lazy:

Predict the MC output for any parameter set.

A strategy

1. Choose a tuning interval for the parameters, then pick random points in parameter space and run the generator with these settings.

2. Interpolate between points $\Rightarrow$ prediction of the MC output at any specific parameter setting.

3. Fit this prediction to data (minimal χ^2).

4. Repeat the fit for different combinations of observables.

5. Choose the nicest set of parameters.

(already described and used in Z. Phys., C 73 (1996) 11-60)

1. Choosing parameters

Pick the parameters you want to tune:

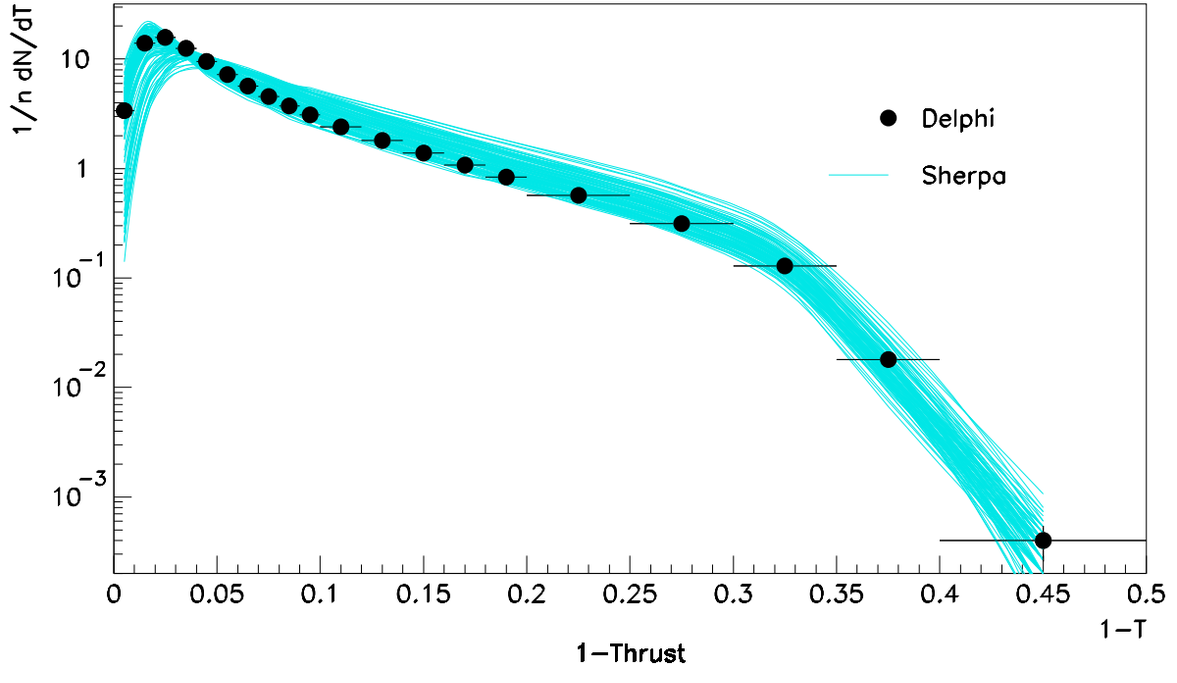
- Tune everything that is important.
- But remember: Each additional parameter adds one dimension to the parameter space.

Define parameter intervals:

- Make the interval large enough so that the result will not be outside.
- But remember: Cutting down 10 intervals by 10 % shrinks the volume of the parameter space by 2/3.

Now pick random points in parameter space and run the generator for each setting.

Calculating observables yields plots like this:



Every line corresponds to a certain parameter setting.

2. Predict the Monte Carlo

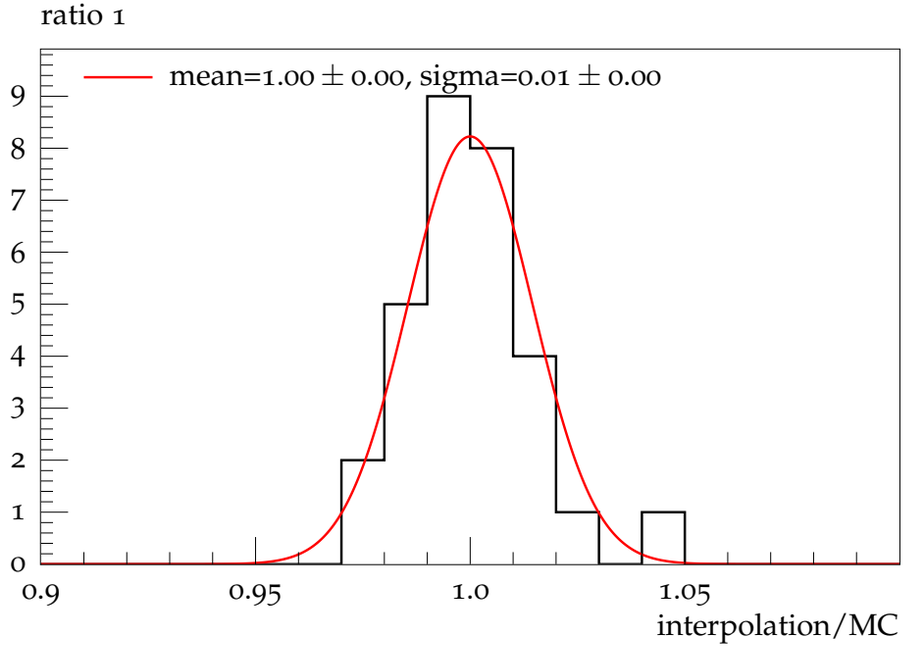
Get a bin by bin prediction for the MC response as function of the parameter set $\vec{p} = (p_1, p_2, \ldots, p_n)$.

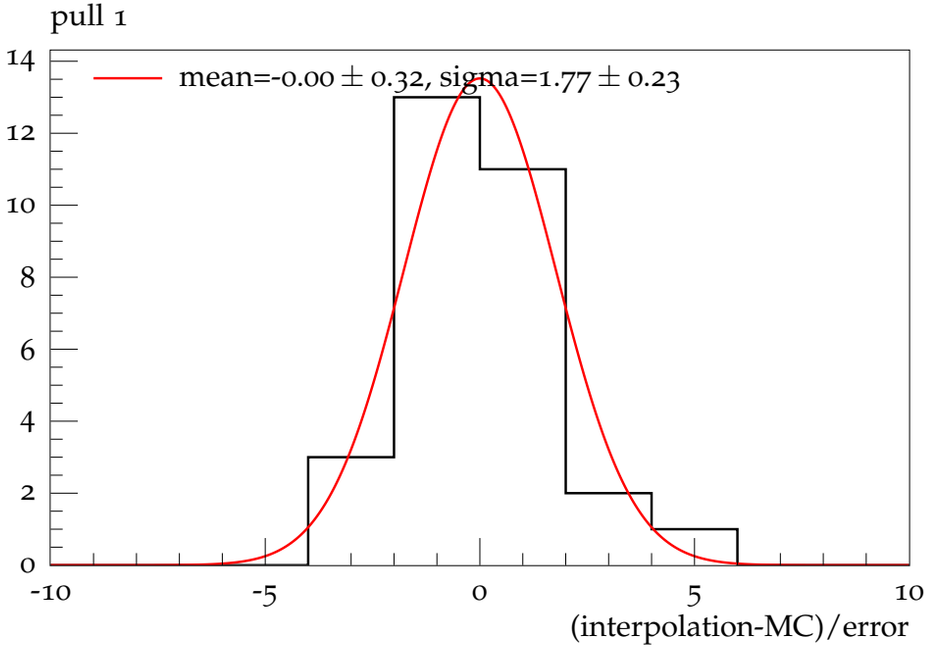
Using a second order polynomial takes the correlations between the parameters into account:

$$X_{\mathrm{MC}}(p_1, p_2, \ldots, p_n) = A_0 + \sum_{i=1}^{n} B_i p_i + \sum_{i=1}^{n} C_i p_i^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} D_{ij} p_i p_j + \cdots$$

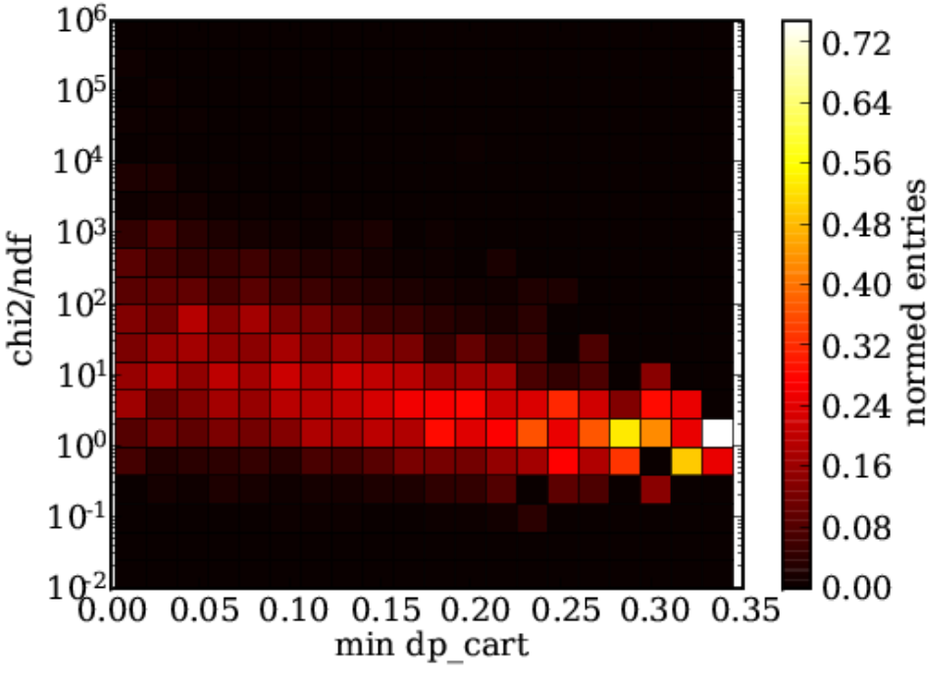
Some interpolation cross-checks

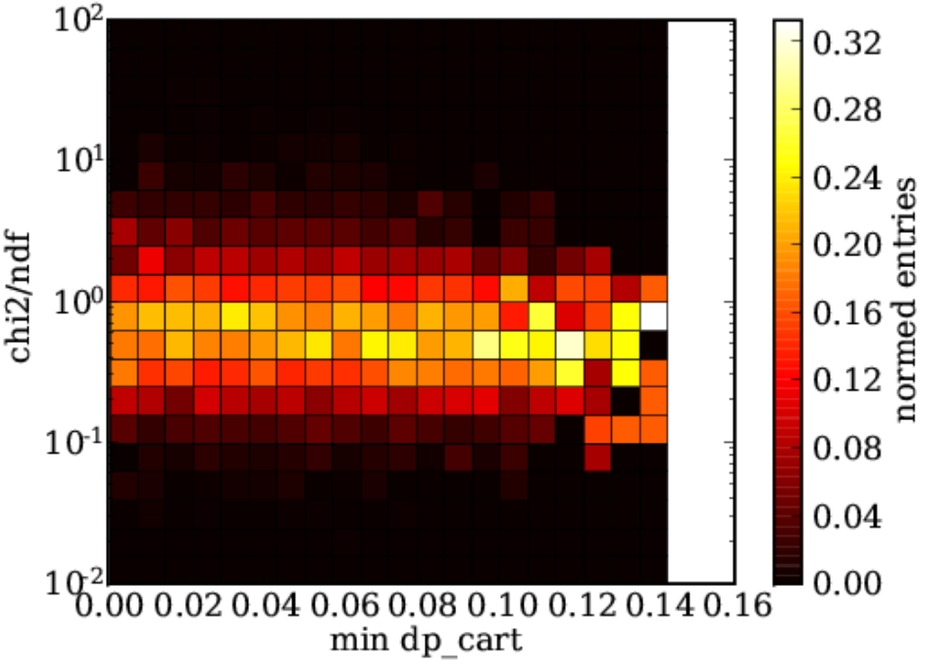
Compare the prediction with the generated MC output, for each bin of each observable:





Check interpolation quality (χ^2) dependence on distance between sampling points:





Control of fit - Grid description

- **Chi2/ndf for polynomial description of parameter space:**

Degreer of polynomial:	2nd	3rd	4th	5th
chi2/ndf for histo 1, bin 1 =	12.46	1.461	1.51	1.71
chi2/ndf for histo 1, bin 2 =	10.76	1.541	1.55	1.53
chi2/ndf for histo 1, bin 3 =	8.057	1.725	1.81	1.44
chi2/ndf for histo 1, bin 4 =	4.194	1.640	1.90	1.42
chi2/ndf for histo 1, bin 5 =	2.021	1.266	1.17	1.37
chi2/ndf for histo 2, bin 1 =	18.52	0.993	0.87	1.31
chi2/ndf for histo 2, bin 2 =	15.57	0.935	0.89	1.26
chi2/ndf for histo 2, bin 3 =	10.39	1.037	1.08	0.92
chi2/ndf for histo 2, bin 4 =	4.439	0.975	1.03	1.07
chi2/ndf for histo 2, bin 5 =	1.950	0.990	0.93	1.13
chi2/ndf for histo 3, bin 1 =	10.91	1.639	1.69	1.76
chi2/ndf for histo 3, bin 2 =	9.129	1.763	1.63	1.64
chi2/ndf for histo 3, bin 3 =	6.594	1.867	1.85	1.22
chi2/ndf for histo 3, bin 4 =	3.016	1.351	1.19	1.49
chi2/ndf for histo 3, bin 5 =	1.426	1.201	1.12	1.21
chi2/ndf for histo 4, bin 1 =	5.219	1.579	1.43	1.30
chi2/ndf for histo 4, bin 2 =	4.454	1.536	1.49	1.25
chi2/ndf for histo 4, bin 3 =	2.738	1.266	1.20	1.32
chi2/ndf for histo 4, bin 4 =	1.651	1.171	1.08	1.24
chi2/ndf for histo 4, bin 5 =	1.036	0.965	1.10	1.08
		1.07		1.514
chi2/ndf for histo 18, bin 1 =	14.7	3.66	2.464	2.864
chi2/ndf for histo 18, bin 2 =	13.0	3.64	2.68	3.120
chi2/ndf for histo 18, bin 3 =	9.68	3.43	2.804	3.228
chi2/ndf for histo 18, bin 4 =	4.77	2.56	2.405	3.255
chi2/ndf for histo 18, bin 5 =	1.44	1.37	1.201	1.268

- **Parameter values from fit to data:**

p1 = 0.372 +/- 0.047	0.310 +/- 0.030	0.284 +/- 0.023	0.293 +/- 0.026
p2 = 0.144 +/- 0.041	0.215 +/- 0.035	0.247 +/- 0.030	0.235 +/- 0.032
p3 = 3. +/- 0.08	3. +/- 0.09	3. +/- 0.05	3. +/- 0.04

- **2nd degree polynomial bad grid description.**
- **For higher orders the final fit is consistent within errors of fit.**

3. Fit the prediction to data

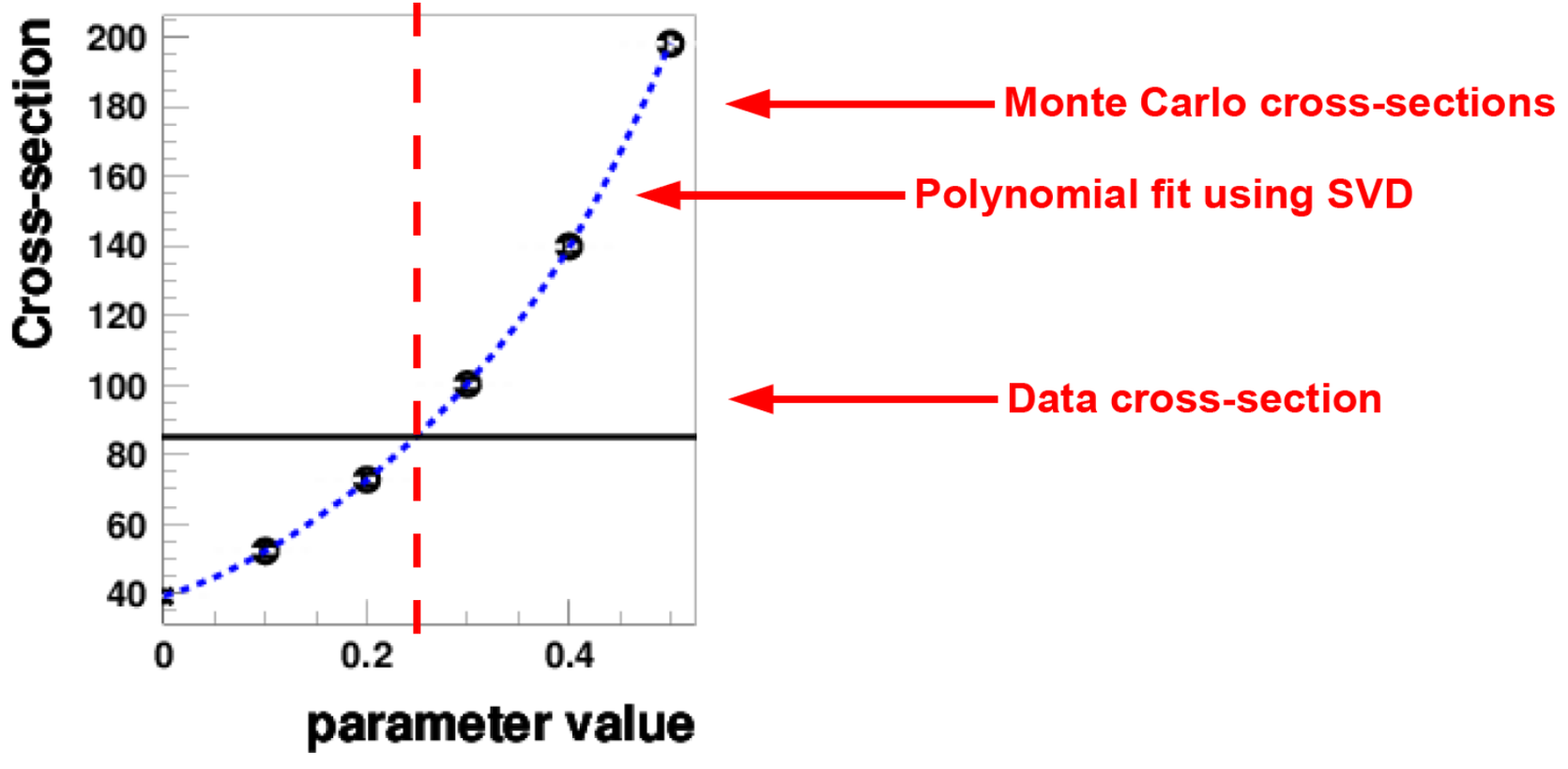
Having A_0, B_i, C_i and D_{ij} we can predict the MC response for any set of parameters very fast. This prediction can be fitted to data, minimising the χ^2:

$$\chi^2(\vec{p}) = \sum_{\text{observables}} \sum_{\text{bins}} \frac{\left(X_{\text{data}} - X_{\text{MC}}(\vec{p})\right)^2}{\sigma^2_{\text{data}} + \sigma^2_{\text{MC}}}$$

Include all the relevant data distributions in the fit!

This fit only takes seconds (as compared to days or weeks for a brute force approach).

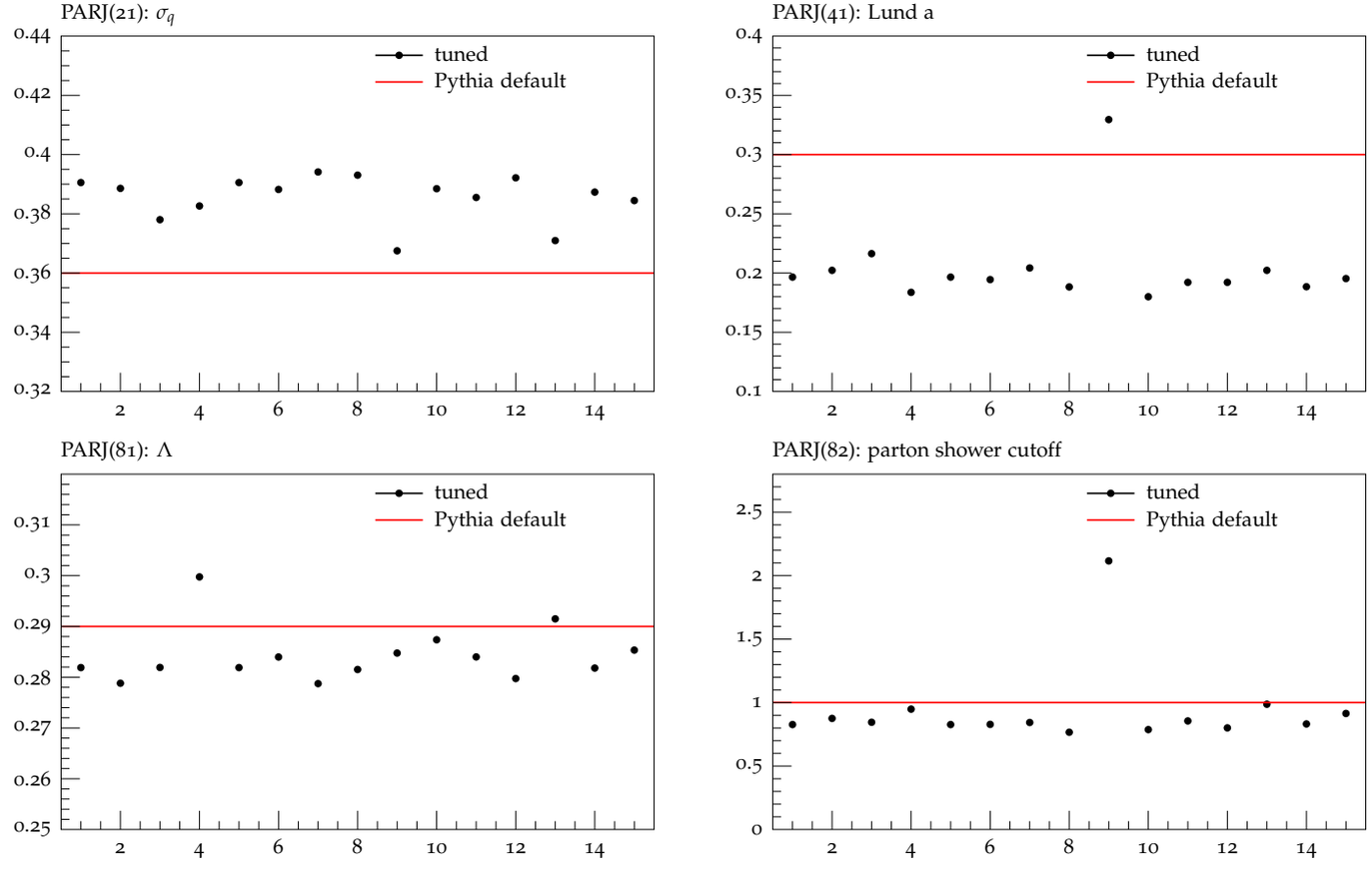
Simplest example: 1 parameter, 1 data bin



4. Use different data sets

Using different combinations of observables yield different optimisation results.

Learn something about correlations and stability of the tuning.



What data should we use . . . ?

It depends . . .

In general: Use observables which are physically related to the parameters you want to tune!

Examples:

For the parton shower use event shapes like Thrust, Sphericity, Planarity, Major, Minor, (differential) jet rates, p_t spectra, N_{ch}, . . .

For hadronisation use identified particle spectra, multiplicities . . .

For PDFs use e. g. F_2, F_2^c, dijets . . .

. . . and what data is available?

- LEP has published *lots* of high precision data.
- There is good data from SLD and the DESY experiments.
- There is very little useful data from the Tevatron!

To compare data to MC either the data needs to be acceptance corrected or the MC needs to be folded with the detector response. *Most of the published Tevatron data satisfies neither condition and can't be compared to anything!*

Implementation – professor

professor is intended to be a general Monte Carlo tuning tool.

Reimplementation of the original tuning strategy described in the Delphi publication, carried out within MCnet.

Tune various generators in a common way by using Rivet/Rivetgun.

Use data from different experiments and accelerators by using Rivet.

Project webpage: http://projects.hepforge.org/professor/

Implementation – PROFFIT

PROFFIT aims at fitting unintegrated PDFs and Monte Carlo predictions from CASCADE.

Implementation is carried out mainly at DESY.

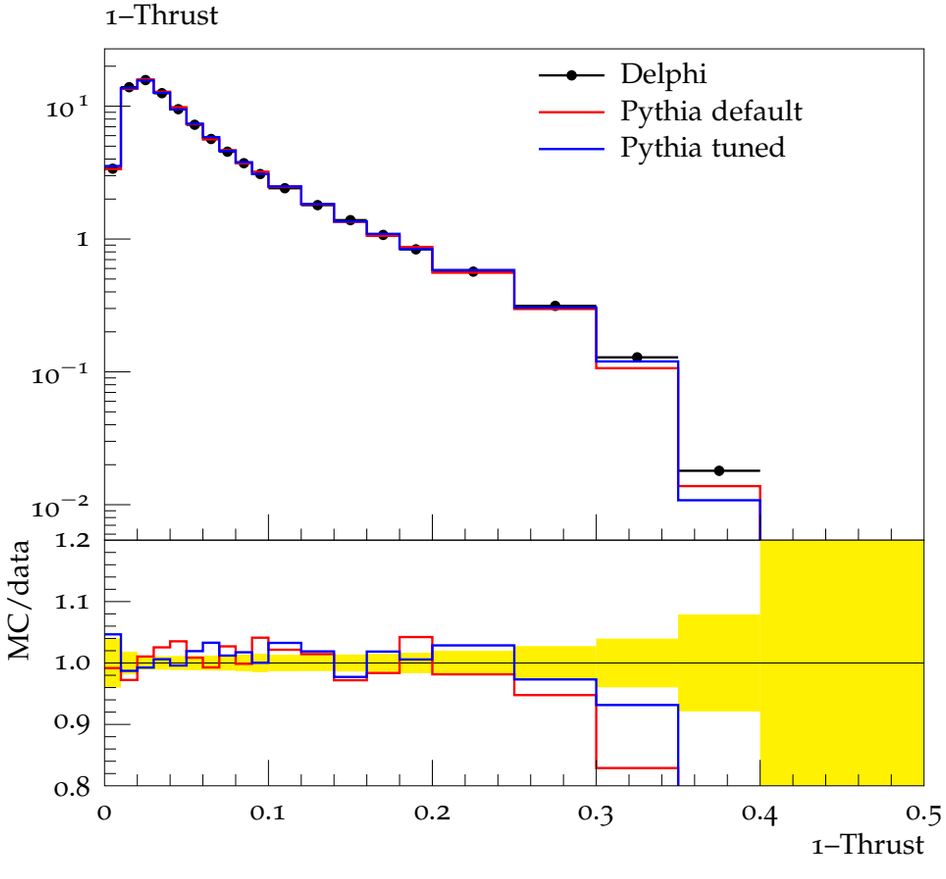
Mainly using data from DIS.

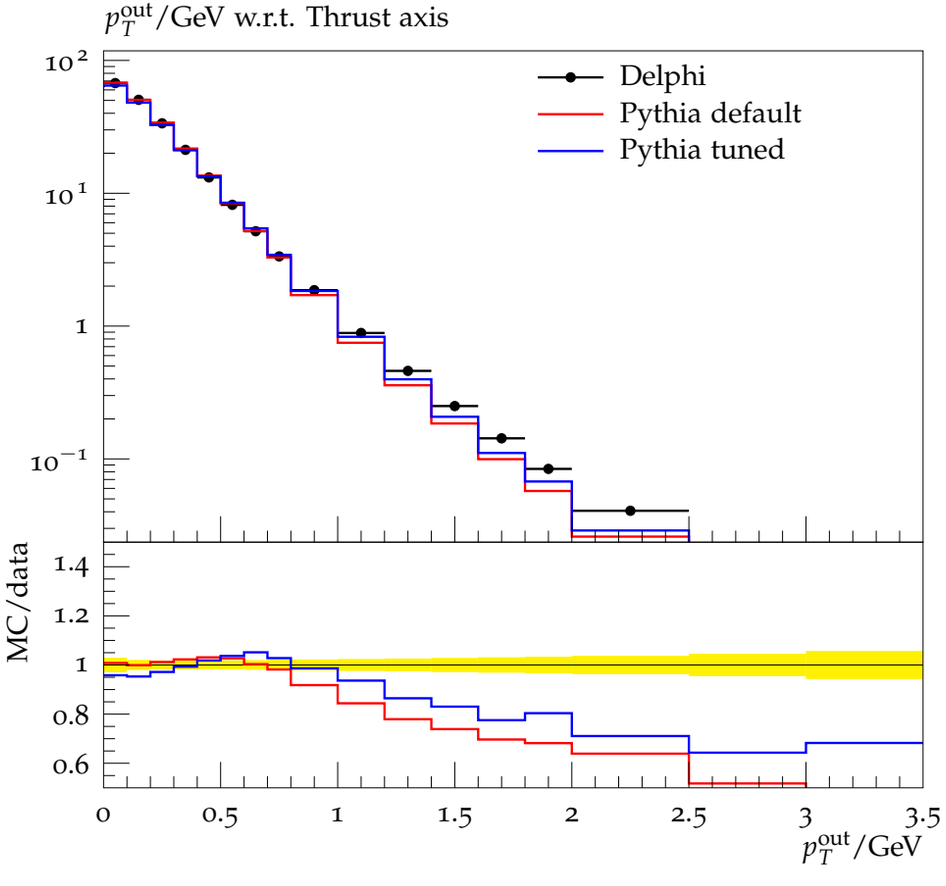
Possible to use up to 5th order polynomial for the interpolation.

Project webpage: http://projects.hepforge.org/proffit/

Example – Pythia 6.4, 4 parameters

Test-tuning of Pythia 6.4: σ_q, Lund a, Λ, PS cutoff





Example – uPDF fit; 3 parameters

Fit of the uPDF parametrization:

$$xA_0(x, k_T, \bar{q}_0) = N \cdot x^{-B} \cdot (1-x)^C \cdot \exp\left(-\frac{(k_T-\mu)^2}{2\sigma^2}\right)$$

N	–	normalization
B	–	small x behaviour
$C = 4$	–	large x behaviour
μ, σ	–	determines the shape of the intrinsic k_T of the gluon below $k_T = 1.2\,\text{GeV}$

Calculation starts at scale $\bar{q}_0$. Evolution done according to CCFM.

N, B, C, μ, and σ are not theoretically calculable, but must be fitted to data.

Example of application

Fit unintegrated gluon density to HERA data

H1 Collab., A. Aktas et al., Eur. Phys. J. C33 (2004) 477

Inclusive Dijet Production at Low x_Bj in DIS

Integrated PDF: DGLAP

LO: **Gluon collinear with proton**

$$k_{t,\mathrm{gluon}} = 0$$

$$\Delta E_{T,jets} = 0 \text{ in HCM}$$

Higher orders:

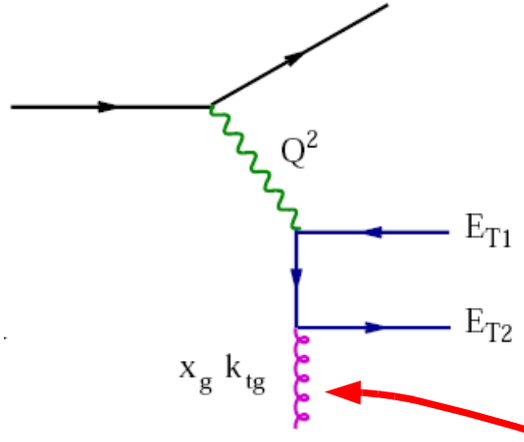
$$k_{t,\mathrm{gluon}} \neq 0$$
$$\Delta E_{T,\mathrm{jets}} \neq 0$$

Unintegrated PDF: CCFM or BFKL

$$k_{t,\mathrm{gluon}} \neq 0$$
$$\Delta E_{t,\mathrm{jets}} \neq 0$$

already at LO

Target hard di-jets.
Dominated by BGF, sensitivity to gluon.



Require $E_{\mathrm{T,\ jet\ 2}} > 5\ \mathrm{GeV}$
$E_{\mathrm{T,\ jet\ 1}} > (5 + \Delta)\ \mathrm{GeV}$

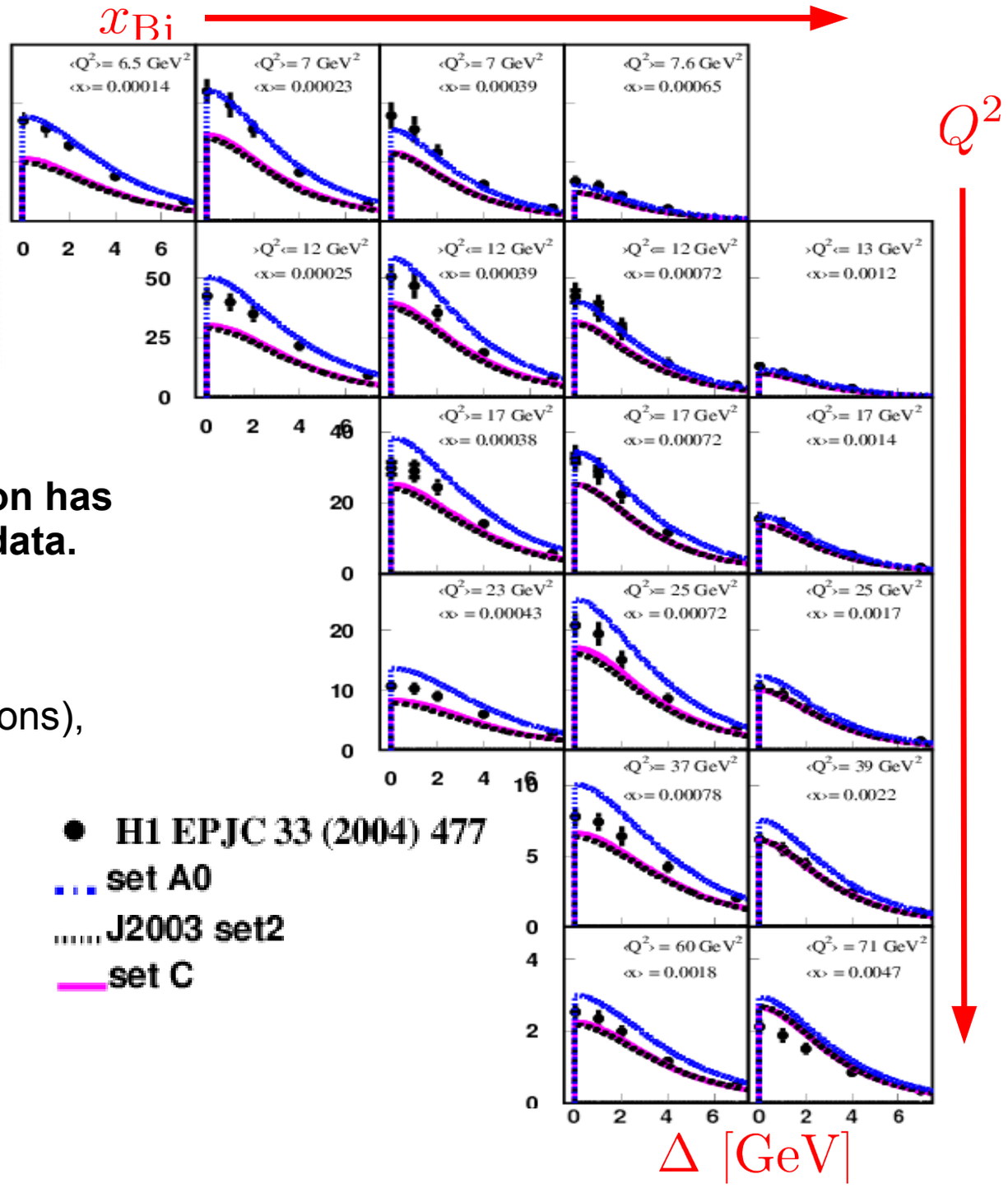
and measure jet cross-section as a function of Δ

Sensitivity to kt of gluon

Di-jet data

$E_{\mathrm{T,\ jet\ 1}} > (5 + \Delta)\ \mathrm{GeV}$

Total dijet cross-section as a function of Δ



Existing CASCADE prediction has some problems describing data.

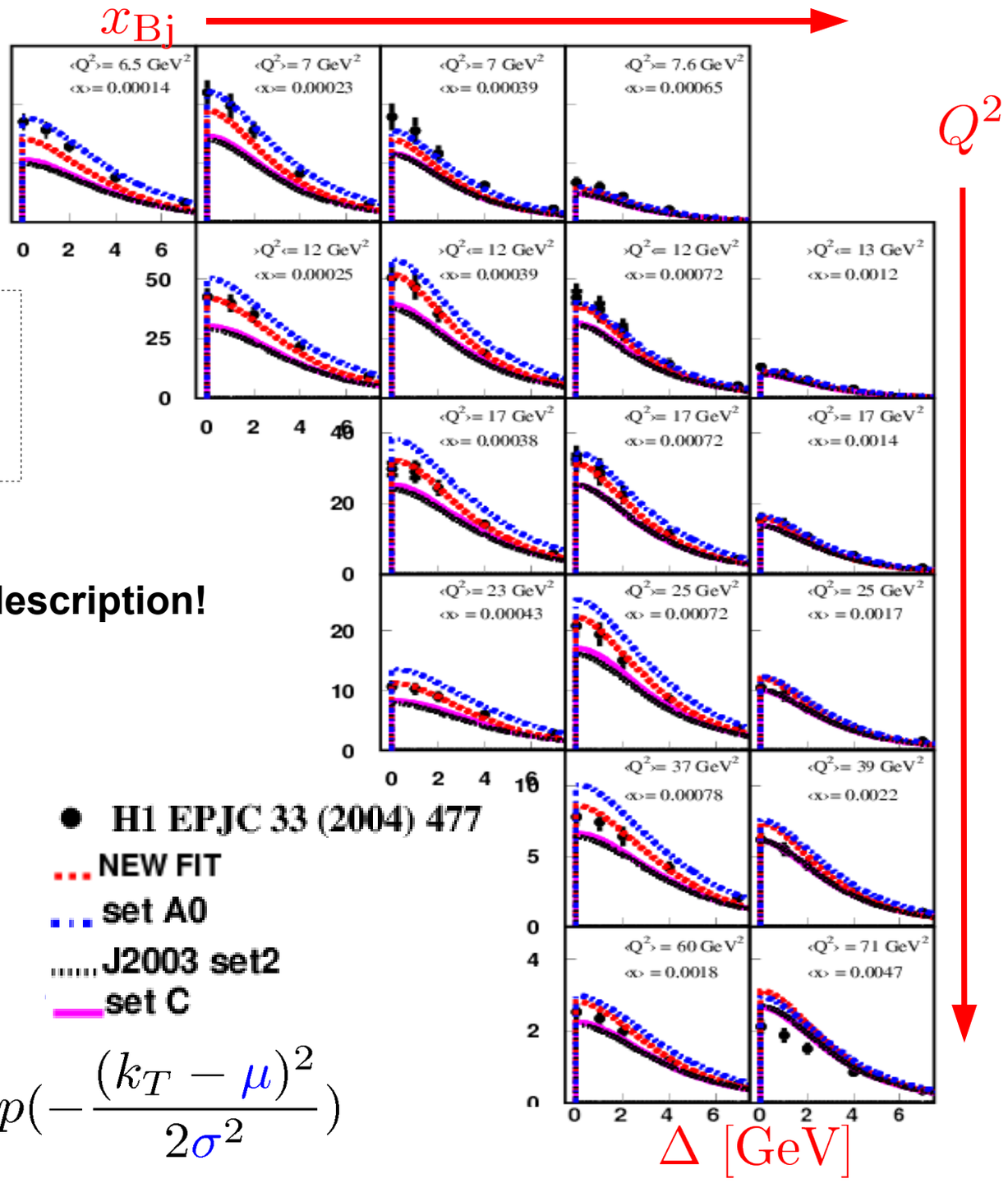
Best is "**set A0**" (determined by fit to proton structure functions), giving a **Chi2/ndf=3.5**

Improve by fitting using PROFFIT...

Di-jet data result

$E_{\mathrm{T,\ jet\ 1}} > (5 + \Delta)\ \mathrm{GeV}$

Total dijet cross-section as a function of Δ



Fitted uPDF improves data description!

N	**0.28 +/- 0.02**
B	**0.25 +/- 0.03**
μ	**3.0 +/- 0.04**
σ	**2, fixed**
Chi2/ndf	**2.01**

- H1 EPJC 33 (2004) 477
- NEW FIT
- set A0
- J2003 set2
- set C

$$N \cdot x^{-B} \cdot (1-x)^{C} \cdot exp(-\frac{(k_T - \mu)^2}{2\sigma^2})$$

Summary

- Monte Carlo event generators are based on phenomenological models.
- The model parameters need tuning to describe the data.
- Parameters are correlated and have to be tuned simultaneously.
- Creating predictive functions for the MC response helps to fit the model parameters to data.
- Good data is crucial for a good tuning.