Forward Modeling for PDF Extraction

Allen Caldwell

Max Planck Institute for Physics

on behalf of

Ritu Aggarwal, SPPU, Pune
Michiel Botje, Nikhef
Francesca Capel*, Max Planck Institute for Physics
Oliver Schulz, Max Planck Institute for Physics

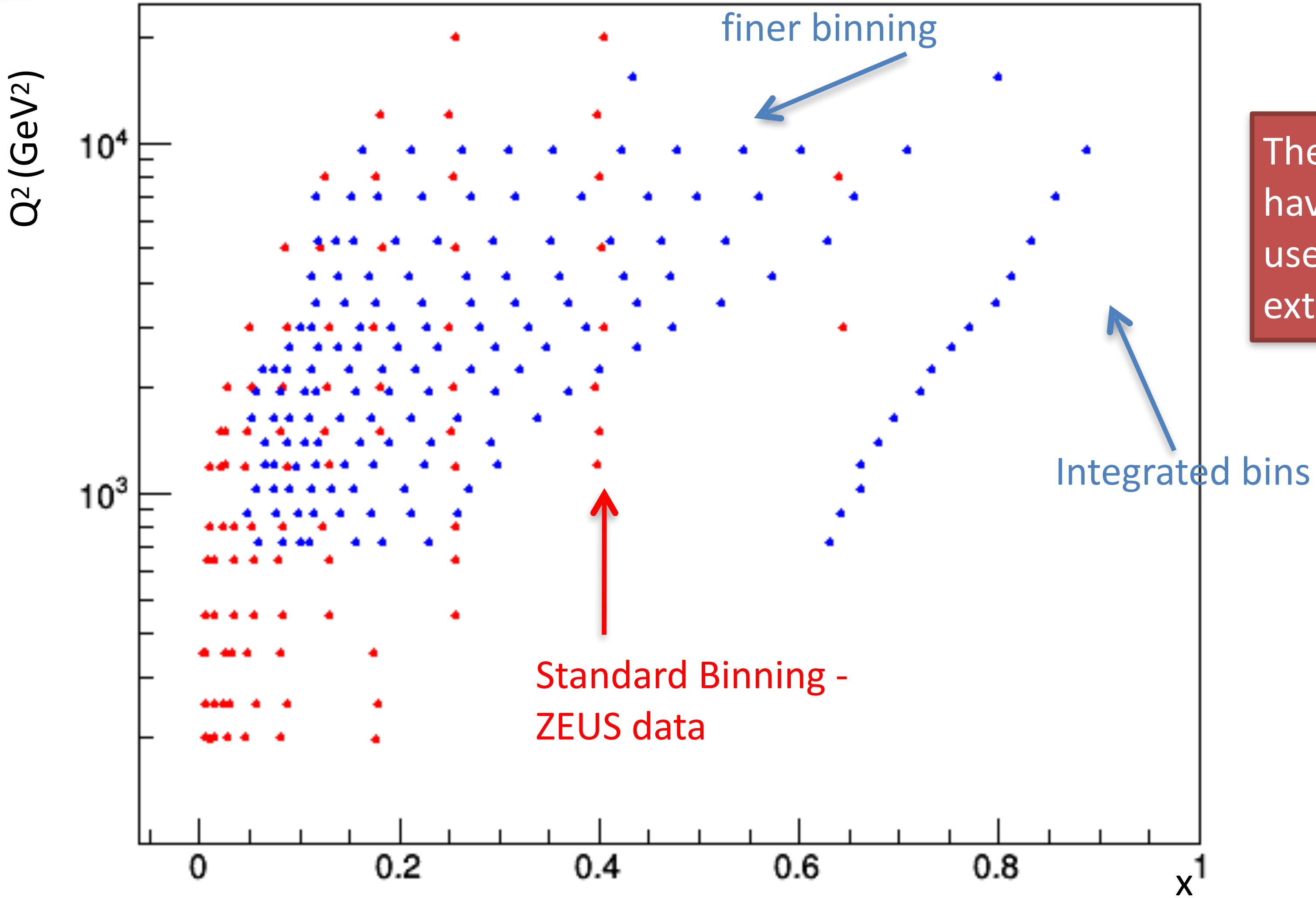
***ORIGINS Data Science Lab**
Technical University, Munich (at the time of this work)

MAX-PLANCK-GESELLSCHAFT

Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

Measurement of neutral current $e^{\pm}p$ cross sections at high Bjorken x with the ZEUS detector

H. Abramowicz *et al.* (ZEUS Collaboration)
Phys. Rev. D **89**, 072007 – Published 8 April 2014



These data have not been used in PDF extractions

Study of proton parton distribution functions at high x using ZEUS data

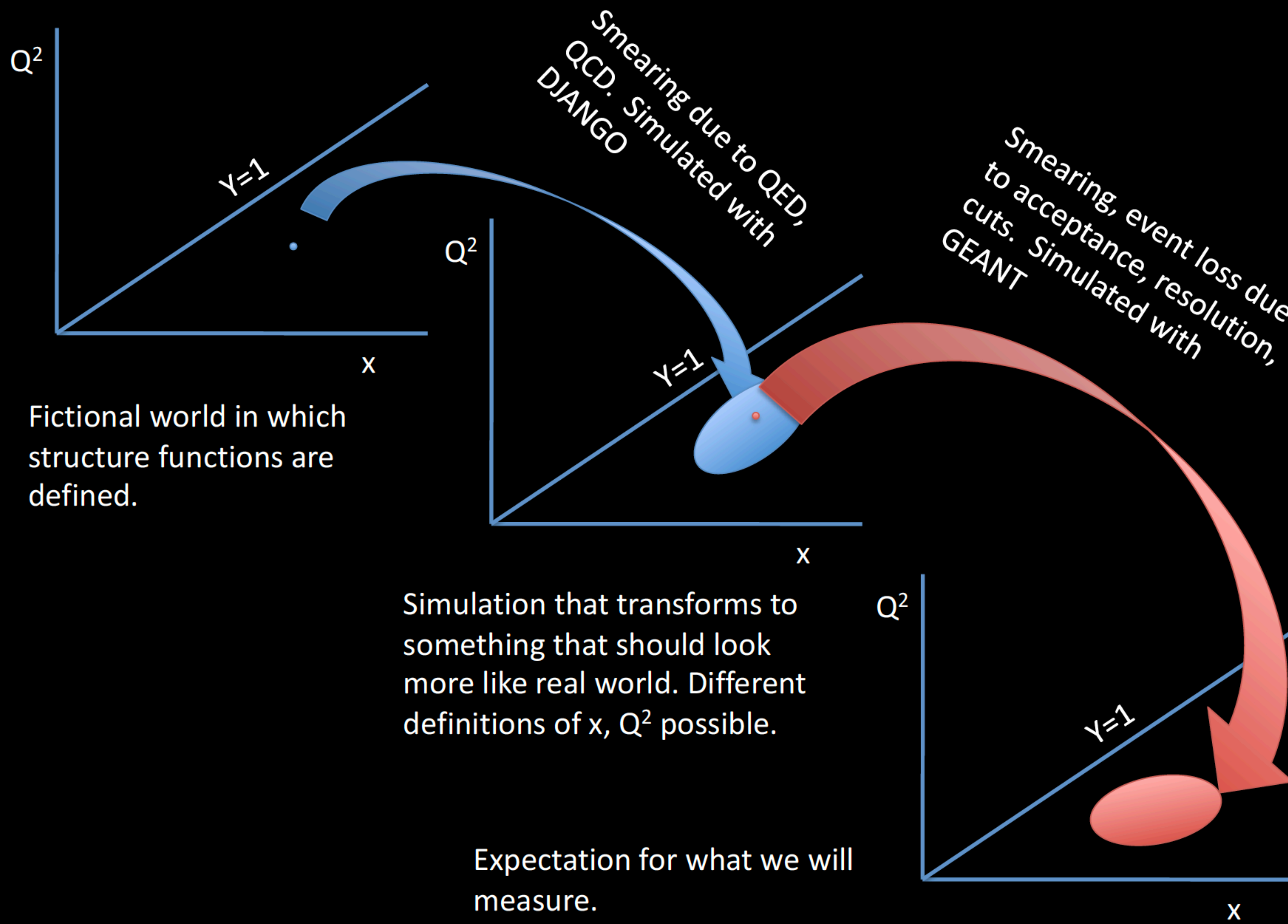
I. Abt *et al.* (ZEUS Collaboration)
Phys. Rev. D **101**, 112009 – Published 26 June 2020

Primary author: **Ritu Aggarwal**

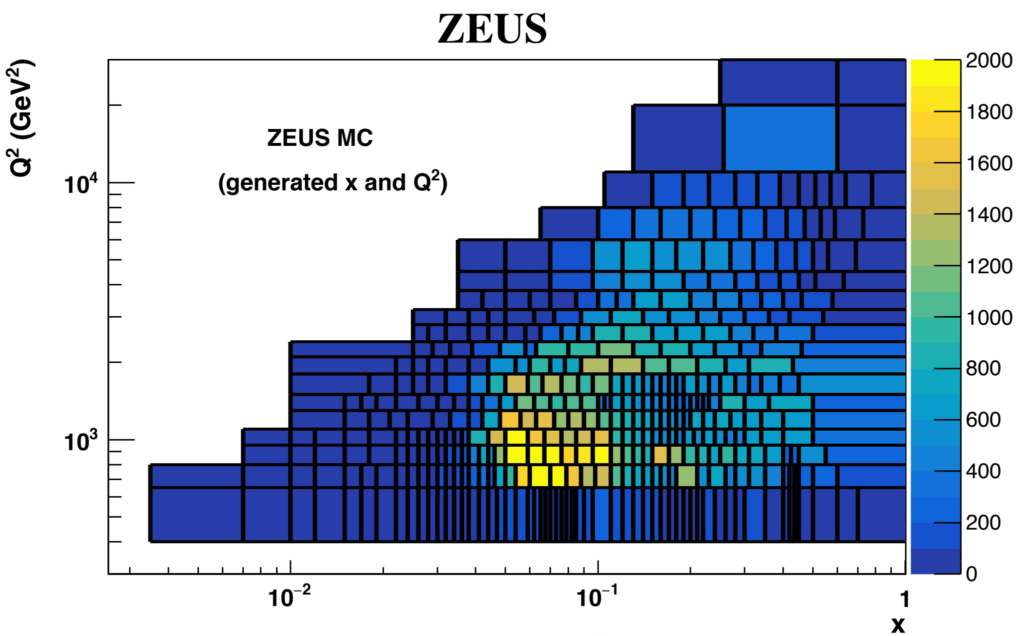
Described how to use a forward modeling for analysis of the data:

Define pdfs -> apply radiative effects
-> predict cross sections
-> apply detector/analysis effects
-> calculate expected number of events
-> calculate a Poisson probability

We are now developing a PDF fitting package to implement this scheme


Q²
Y=1
x
Fictional world in which structure functions are defined.
Smearing due to QED, QCD. Simulated with DJANGO
Q²
Y=1
x
Simulation that transforms to something that should look more like real world. Different definitions of x, Q² possible.
Smearing, event loss due to acceptance, resolution, cuts. Simulated with GEANT
Q²
Y=1
x
Expectation for what we will measure.

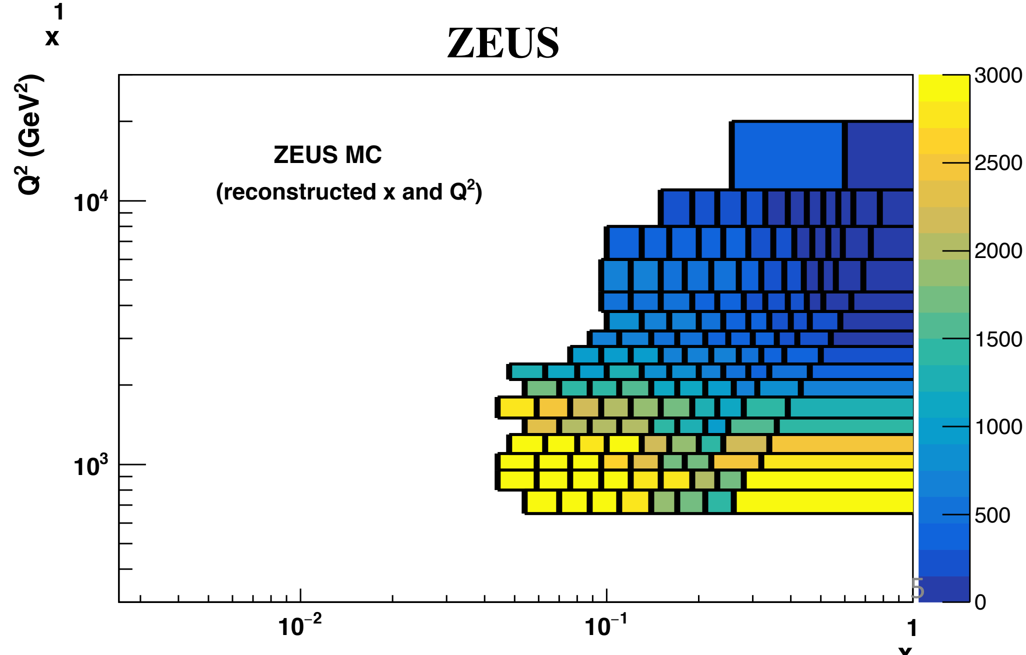
Transfer Matrices



generated variables/bins.

Separate transfer matrices exist for producing radiative cross sections and detector/ analysis effects.

transfer matrix



Reconstructed variables/bins

Developed by **R. Aggarwal**

Procedure

- PDFs defined at a high scale: $Q_0^2 = 100\ \mathrm{GeV}^2$ in the Fixed Flavor number scheme (5 quarks)

- PDFs are evolved at NNLO using QCDNUM to cover the full range of the data

- Structure functions are computed with QCDNUM and represented by cubic splines. These are then used to form the differential cross section, which is also splined. This allows for a fast integration of the cross sections.

- The predictions at the observed level are then calculated using the transfer matrices

expected counts at generator level

$$\nu_j = (1 + 0.018 \cdot \beta_0^{+-}) \left[\sum_i \nu_i \cdot (a_{ij} + \sum_k \beta_k \delta_{ij}^k) \right]$$

normalization uncertainty

transfer matrix

systematic variations

$\beta's$ are Unit Normal distributed nuisance parameters

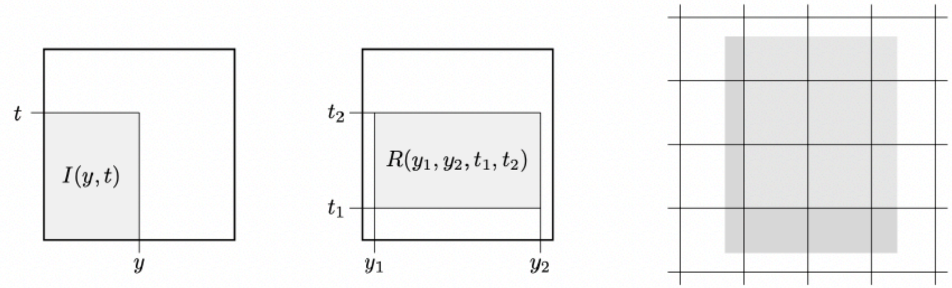
The probability of observing the data is then calculated using the Poisson distribution

QCDNUM/SPLINT

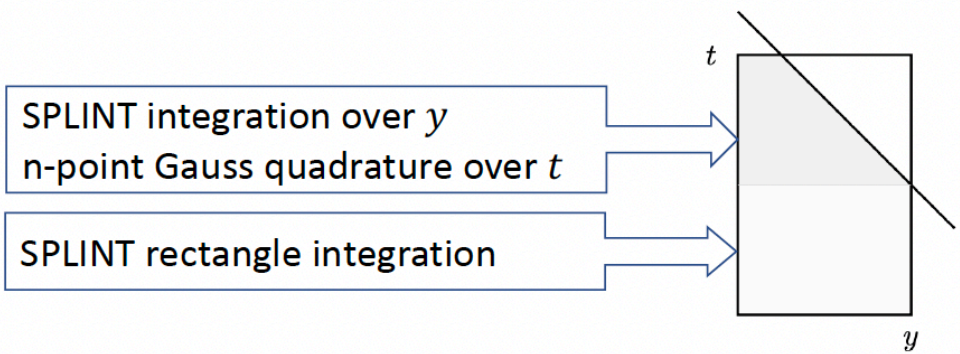
SPLINT 2-dimensional integration

- Spline in $y = -\log x$ and $t = \log \mu^2$ introduces Jacobian $e^{-y} e^{t}$ in the integral

- Splint has very fast routines to integrate over rectangles



- Can now also handle kinematic limit



Quite some mathematical detail: see Appendix A of the SPLINT write-up

QCDNUM/SPLINT

Here is the complete timing summary

	n_x	n_q	t [ms]
Evolution	100	50	3.6
6 Stf splines	22	7	2.9
Xsec spline	100	25	2.2
Integration			0.8

Michiel Botje

The speedup is almost three orders of magnitude compared to initial attempts using standard numerical integration techniques with relative accuracy better than $5 \cdot 10^{-4}$

Allows to run MCMC chain with many iterations in reasonable time.

QCDNUM written in Fortran, with a C++ interface. A QCDNUM interface to the Julia programming language is now also available.

The BAT.jl package is written in the Julia Programming language.
Interfacing to MCMC package and development of PDF fitting code -> **Francesca Capel**

Bayesian Analysis Toolkit

docs stable | docs dev | license MIT | CI passing | codecov 59% | DOI 10.5281/zenodo.2587213

Welcome to BAT, a Bayesian analysis toolkit in Julia.

BAT.jl currently includes: (among other things)

- Metropolis-Hastings MCMC sampling, Hamiltonian MC, Nested Sampling
- Adaptive Harmonic Mean Integration (AHMI)
- Plotting recipes for MCMC samples and statistics

Additional sampling algorithms and other features are in preparation.

BAT.jl originated as a rewrite/redesign of BAT, the Bayesian Analysis Toolkit in C++. BAT.jl now offer a different set of functionality and a wider variety of algorithms than it's C++ predecessor.

Project lead & primary author: **Oliver Schulz**

A first try

$$Q_0^2 = 100\ \mathrm{GeV}^2$$

$$\sum_i \int_0^1 x f_i(x) dx = \sum_i \Delta_i = 1$$

Densities & evolution in FFN (5) scheme & NNLO

$$\int_0^1 u(x) - \bar{u}(x) dx = 2$$

$$\int_0^1 d(x) - \bar{d}(x) dx = 1$$

$$\int_0^1 f(x) - \bar{f}(x) dx = 0$$

$$f \neq u, d, g$$

Parametrizations

$$xu_V(x) = xu(x) - x\bar{u}(x) = A_u x^{\lambda_u}(1-x)^{K_u}$$

$$xd_V(x) = xd(x) - x\bar{d}(x) = A_d x^{\lambda_d}(1-x)^{K_d}$$

$$x\bar{u}(x) = A_{\bar{u}} x^{\lambda_q}(1-x)^{K_q}$$
$$x\bar{d}(x) = A_{\bar{d}} x^{\lambda_q}(1-x)^{K_q}$$
$$xg(x) = A_{g1} x^{\lambda_{g1}}(1-x)^{K_g} + A_{g2} x^{\lambda_{g2}}(1-x)^{K_q}$$
$$xs(x) = x\bar{s}(x) = A_s x^{\lambda_q}(1-x)^{K_q}$$
$$xc(x) = x\bar{c}(x) = A_c x^{\lambda_q}(1-x)^{K_q}$$
$$xb(x) = x\bar{b}(x) = A_b x^{\lambda_q}(1-x)^{K_q}$$

Fit parameters are

$$\Delta_i's, K_u, K_d, \lambda_{g1}, \lambda_{g2}, K_g, \lambda_q + \beta$$

β are nuisance parameters (systematics)

$K_q = 5$ fixed (pdf zero as $x \to 1$)

2 free parameters for data normalization

A first try

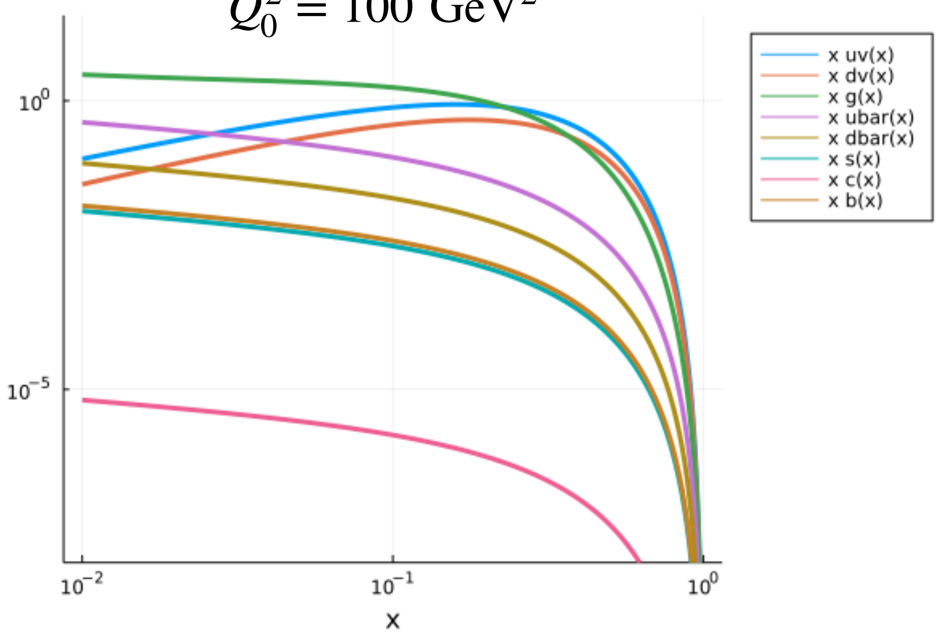
Parametrizations

$$xu_V(x) = xu(x) - x\bar{u}(x) = A_u x^{\lambda_u}(1-x)^{K_u}$$

$$xd_V(x) = xd(x) - x\bar{d}(x) = A_d x^{\lambda_d}(1-x)^{K_d}$$

$$x\bar{u}(x) = A_{\bar{u}} x^{\lambda_q}(1-x)^{K_{\bar{q}}}$$
$$x\bar{d}(x) = A_{\bar{d}} x^{\lambda_q}(1-x)^{K_{\bar{q}}}$$
$$xg(x) = A_{g1} x^{\lambda_{g1}}(1-x)^{K_g} + A_{g2} x^{\lambda_{g2}}(1-x)^{K_{\bar{q}}}$$
$$xs(x) = x\bar{s}(x) = A_s x^{\lambda_q}(1-x)^{K_{\bar{q}}}$$
$$xc(x) = x\bar{c}(x) = A_c x^{\lambda_q}(1-x)^{K_{\bar{q}}}$$
$$xb(x) = x\bar{b}(x) = A_b x^{\lambda_q}(1-x)^{K_{\bar{q}}}$$

$Q_0^2 = 100\ \mathrm{GeV}^2$



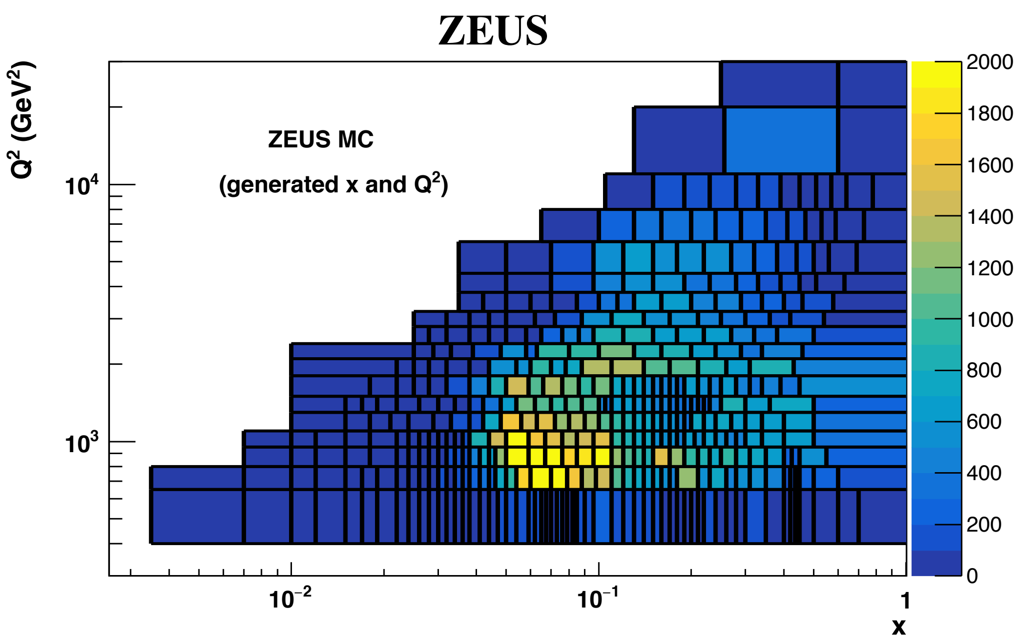
Input values

$$\begin{aligned}
\Delta_{uv} &= 0.28, \quad K_u = 6.0\\
\Delta_{dv} &= 0.15, \quad K_d = 6.0\\
\Delta_{g1} &= 0.29, \quad K_g = 7.0\\
\Delta_{g2} &= 0.18\\
\Delta_{usea} &= 0.073\\
\Delta_{dsea} &= 0.014\\
\Delta_{s} &= 0.002\\
\Delta_{c} &= 0.000001\\
\Delta_{b} &= 0.003
\end{aligned}$$

$$\lambda_{g1} = 1.5, \quad \lambda_{g2} = -0.4, \quad \lambda_q = -0.4$$

$\Delta's$ randomly generated from prior

A first try

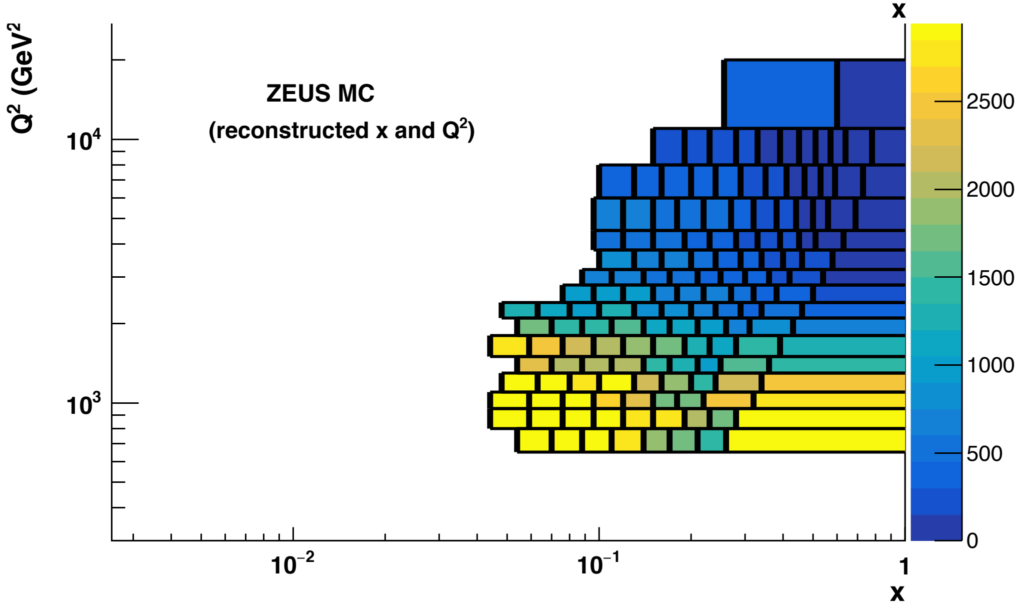


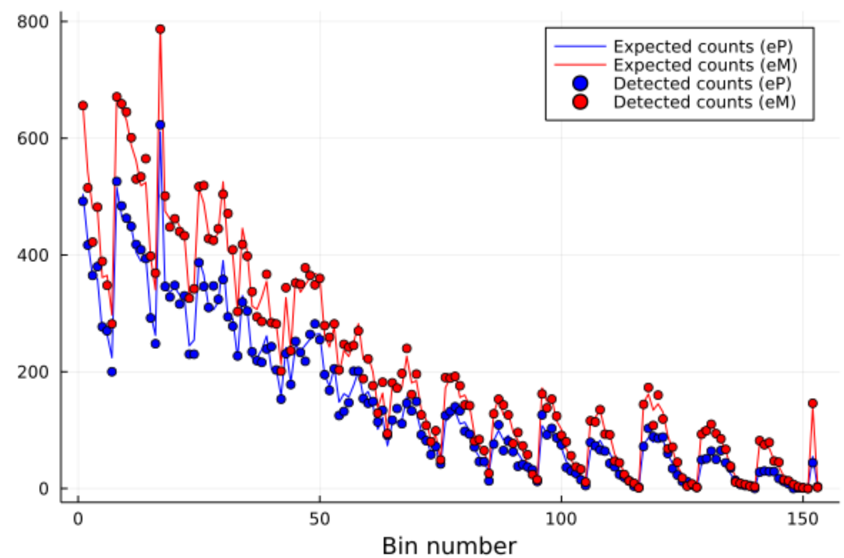
QCDNUM evolves PDFs to cover grid:

SPLINT package gives integrated cross sections in bins

transfer matrix used to get expected numbers of events in bins of observed quantities.

Poisson generated number of events.



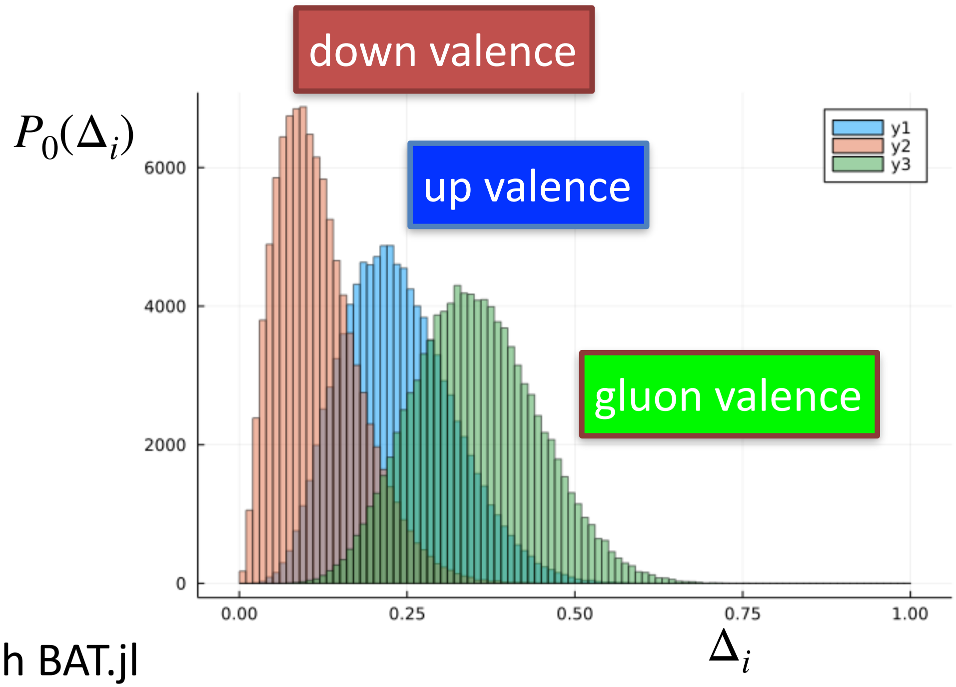


A first try

Priors

Δ = Dirichlet([6., 3., 9., 4., 2.,1., 0.2, 0.2, 0.1]),
K_u= Uniform(3., 9.),
K_d= Uniform(3., 9.),
λ_{g1} = Uniform(1., 2.),
λ_{g2}= Uniform(-0.5, -0.1),
K_g= Uniform(3., 9.),
λ_q = Uniform(-0.5, -0.1),
β_0^+ = Truncated(Normal(0, 1), -5, 5),
β_0^-= Truncated(Normal(0, 1), -5, 5),

Some of the $\Delta's$



Markov Chain MC used to fit simulated data with BAT.jl

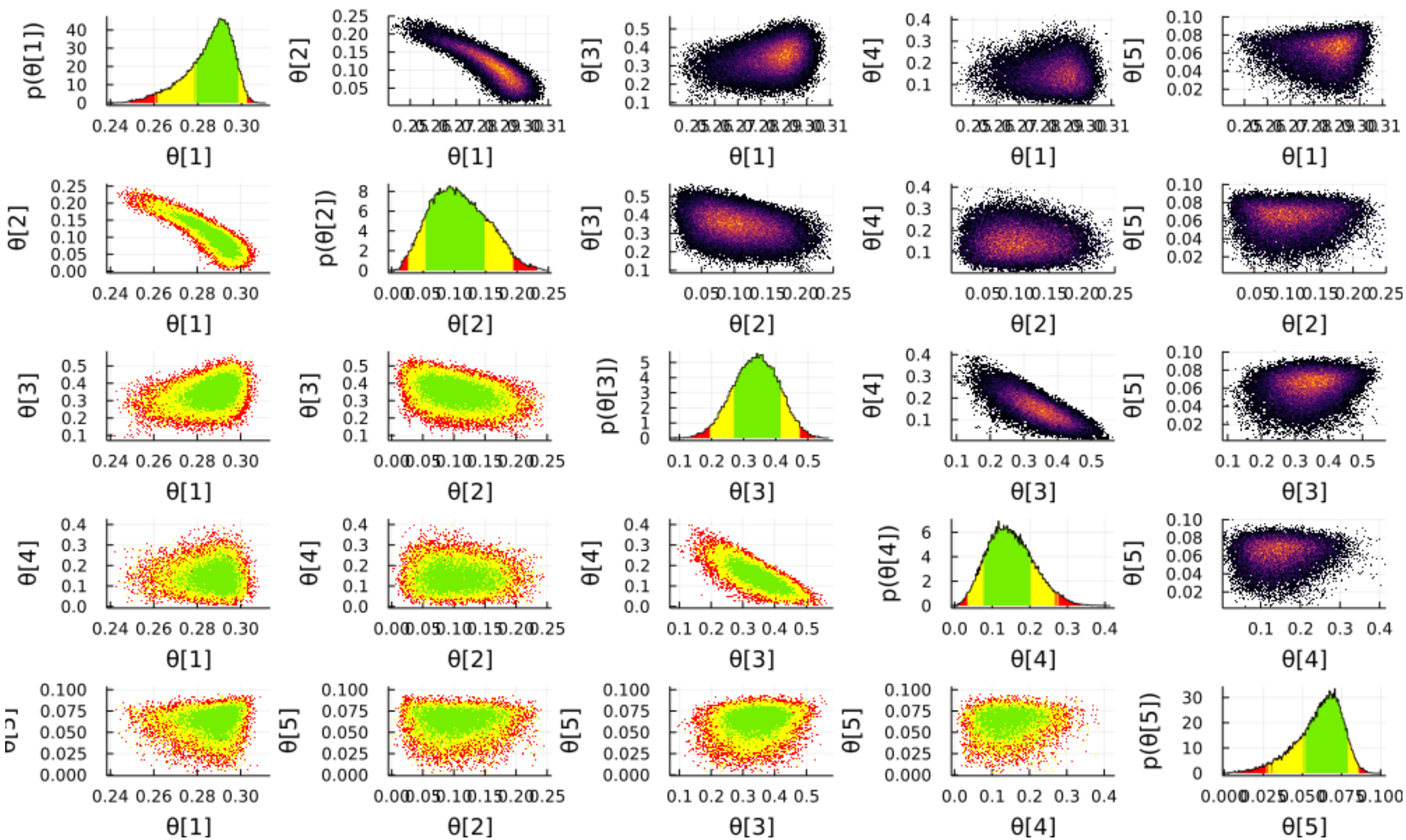
$$P(\mathbf{\Delta}, K_u, K_d, \lambda_{g1}, \lambda_{g2}, K_g, \lambda_q \,|\, D) \propto P(D \,|\, \mathbf{\Delta}, K_u, K_d, \lambda_{g1}, \lambda_{g2}, K_g, \lambda_q) P_0(\mathbf{\Delta}, K_u, K_d, \lambda_{g1}, \lambda_{g2}, K_g, \lambda_q)$$

Some results …
Fitting code: **F. Capel** implemented fitting model, BAT.jl **O. Schulz** et al.)

MCMC Output

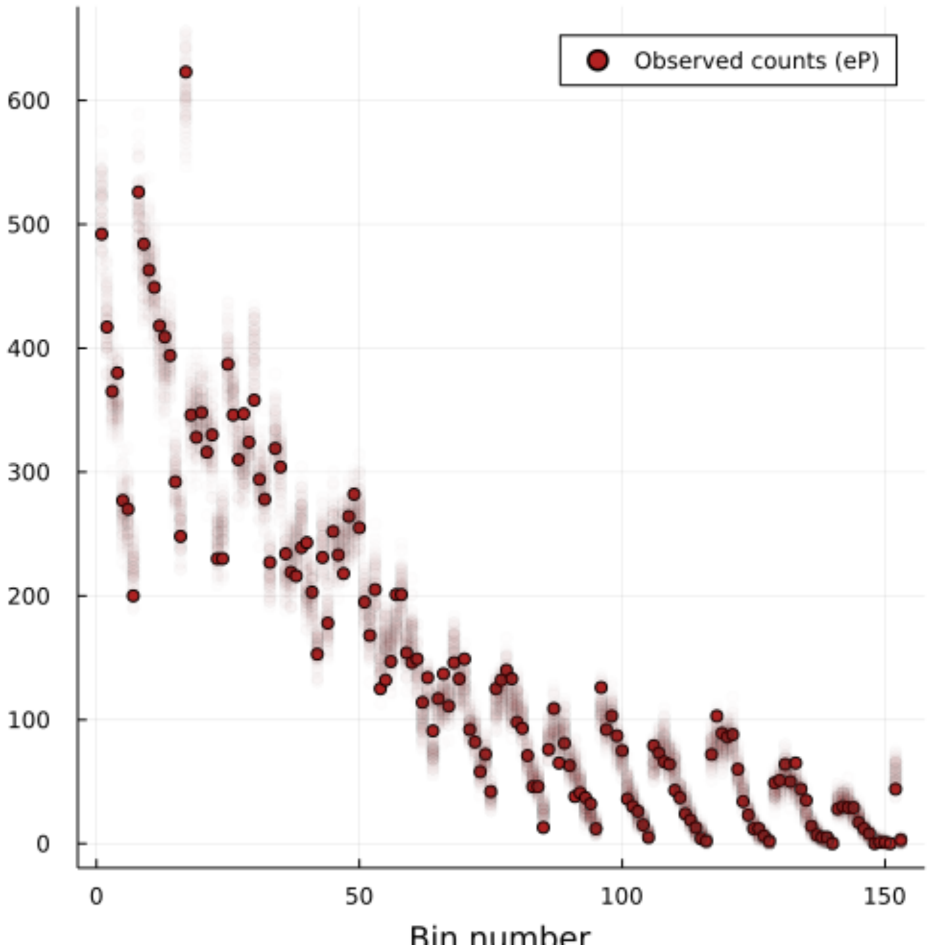
Output is $\{\boldsymbol{\Delta}, K_u, K_d, \lambda_{g1}, \lambda_{g2}, K_g, \lambda_q, \beta\}$ distributed $\propto P(\boldsymbol{\Delta}, K_u, K_d, \lambda_{g1}, \lambda_{g2}, K_g, \lambda_q, \beta \,|\, D)$.

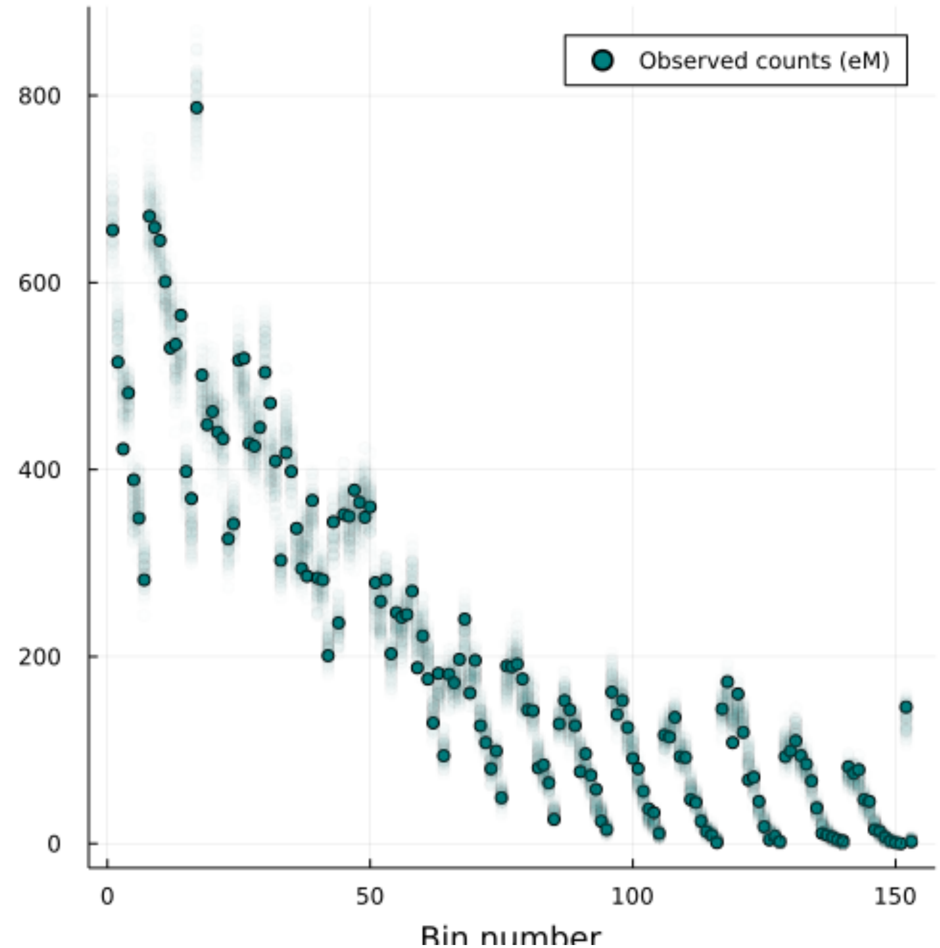
BAT.jl outputs all 1,2D marginalized distributions. A small subset of possible plots.



Data/Model Posterior Check

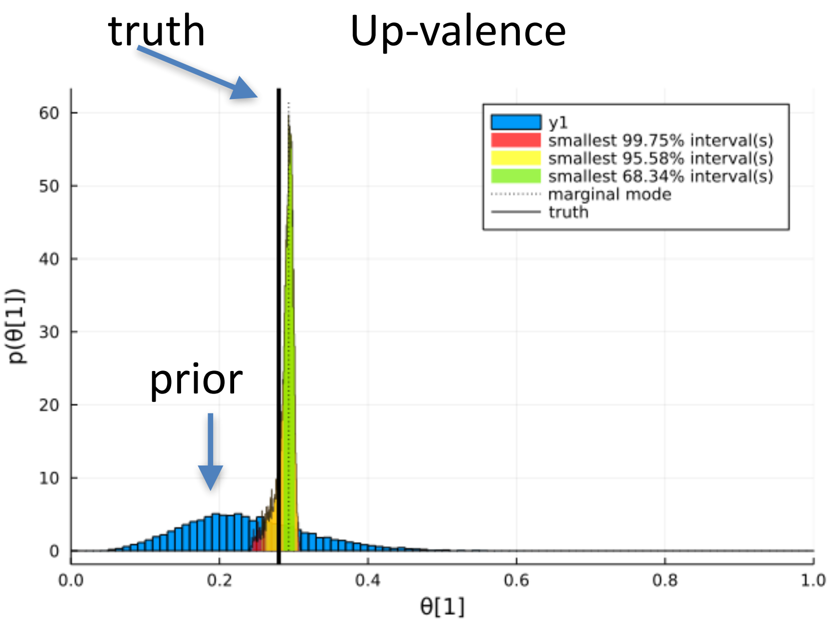
Predictions for event numbers from posterior parameter sets



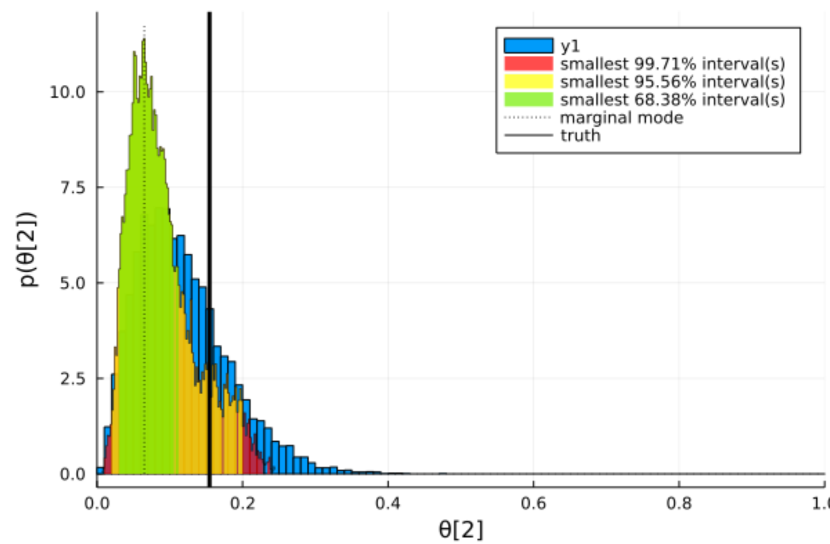


Momenta

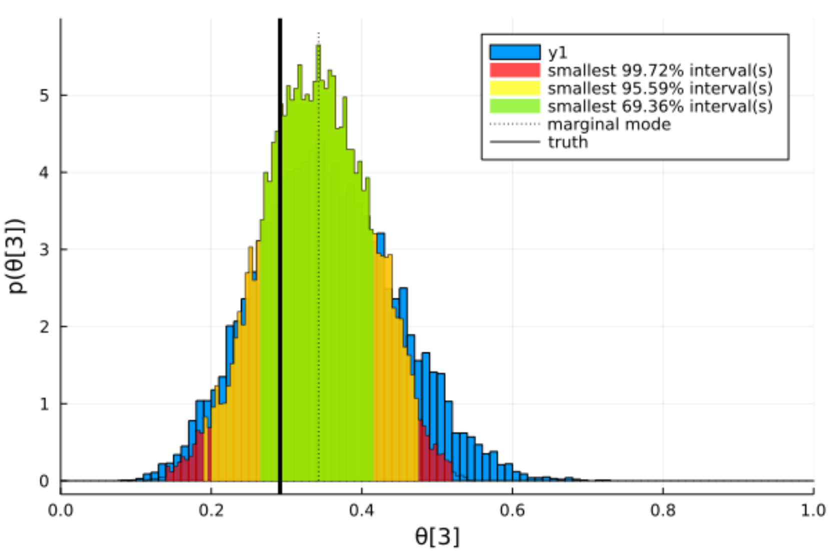
Up-valence



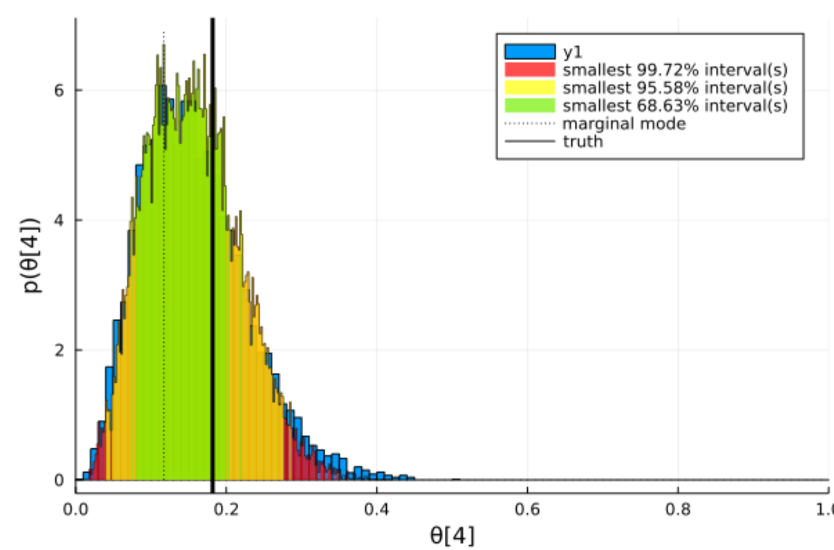
Down-valence



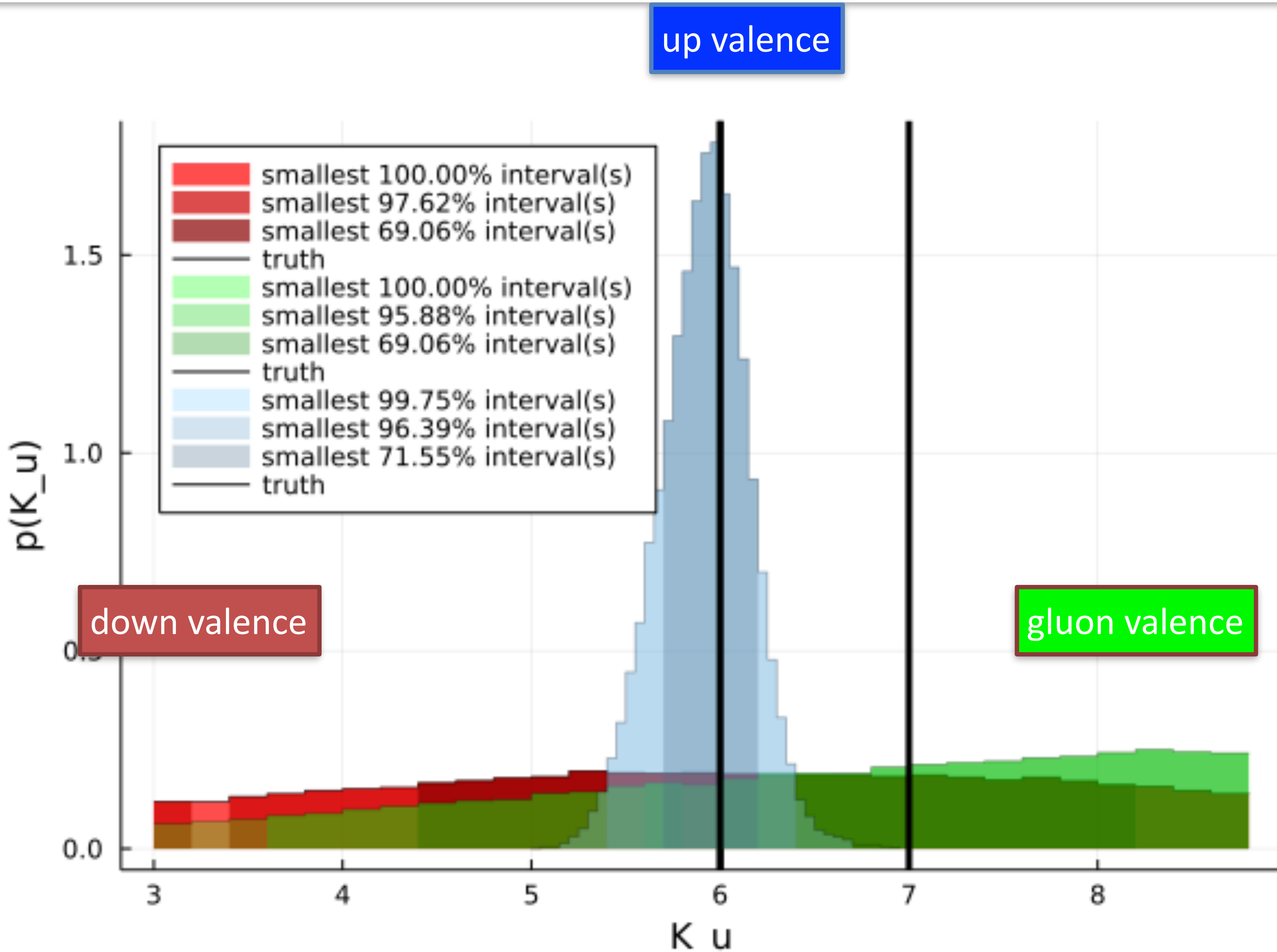
Glue-valence



Glue-sea

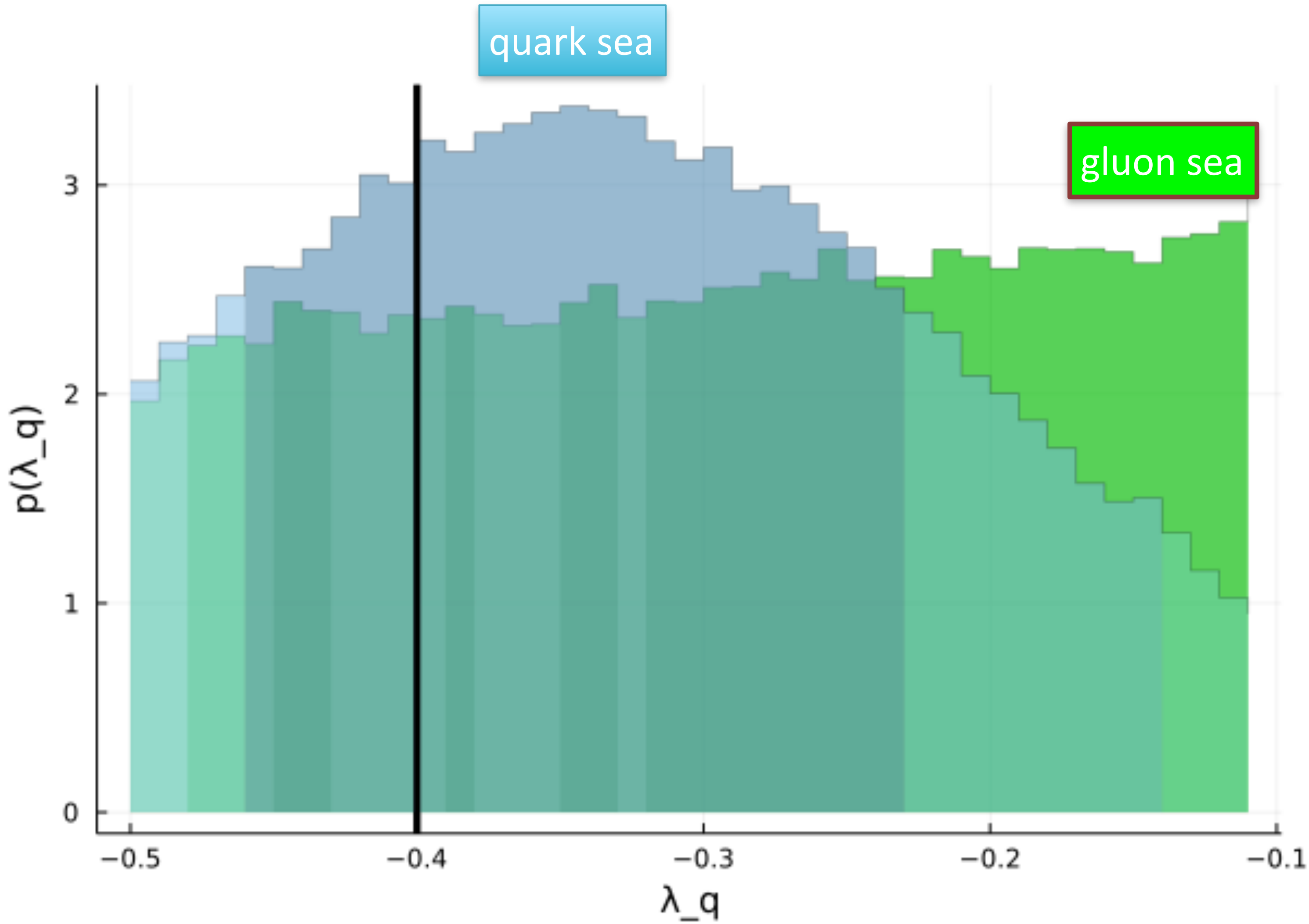


Shape Parameters



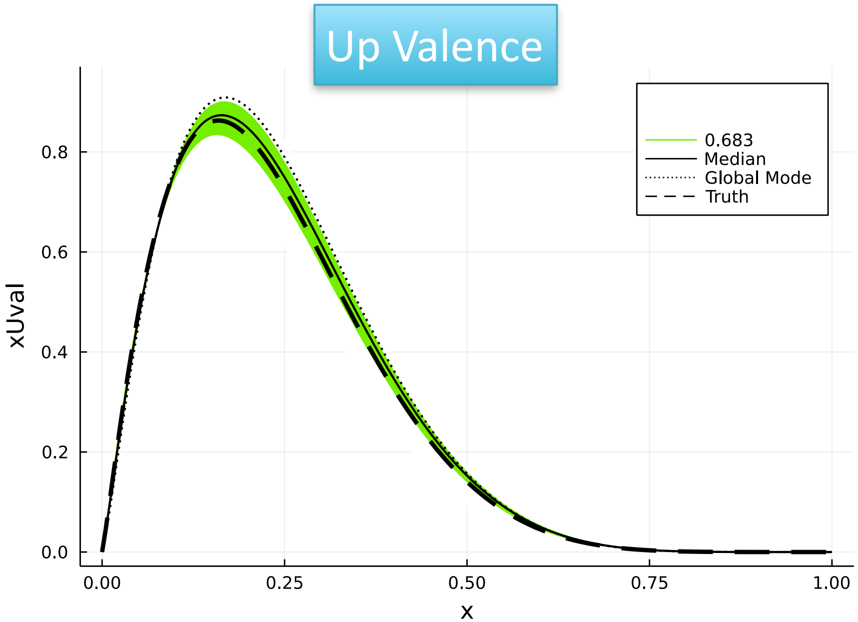
Shape (and momentum) of up-valence well constrained. Others weakly constrained.

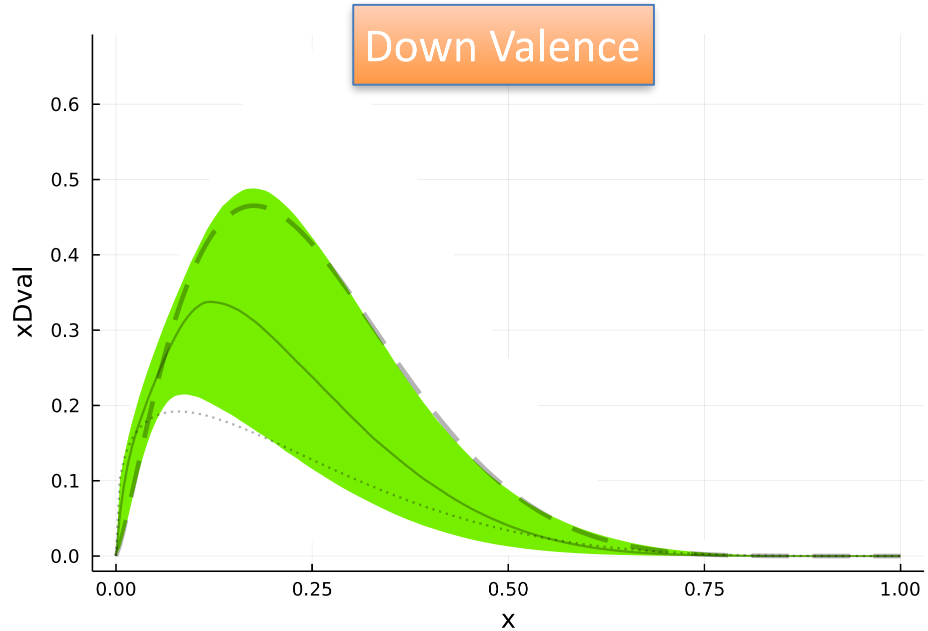
Shape Parameters

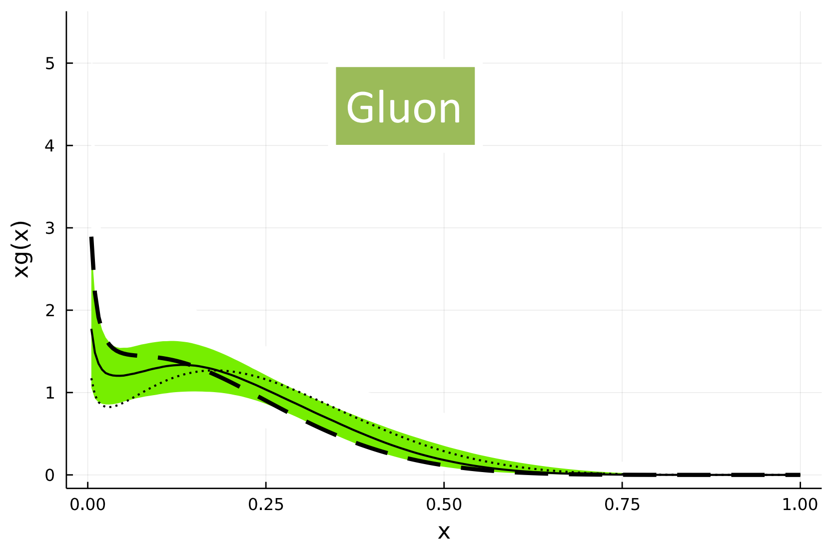


weakly informative data on the sea distributions

Parton Densities







Summary

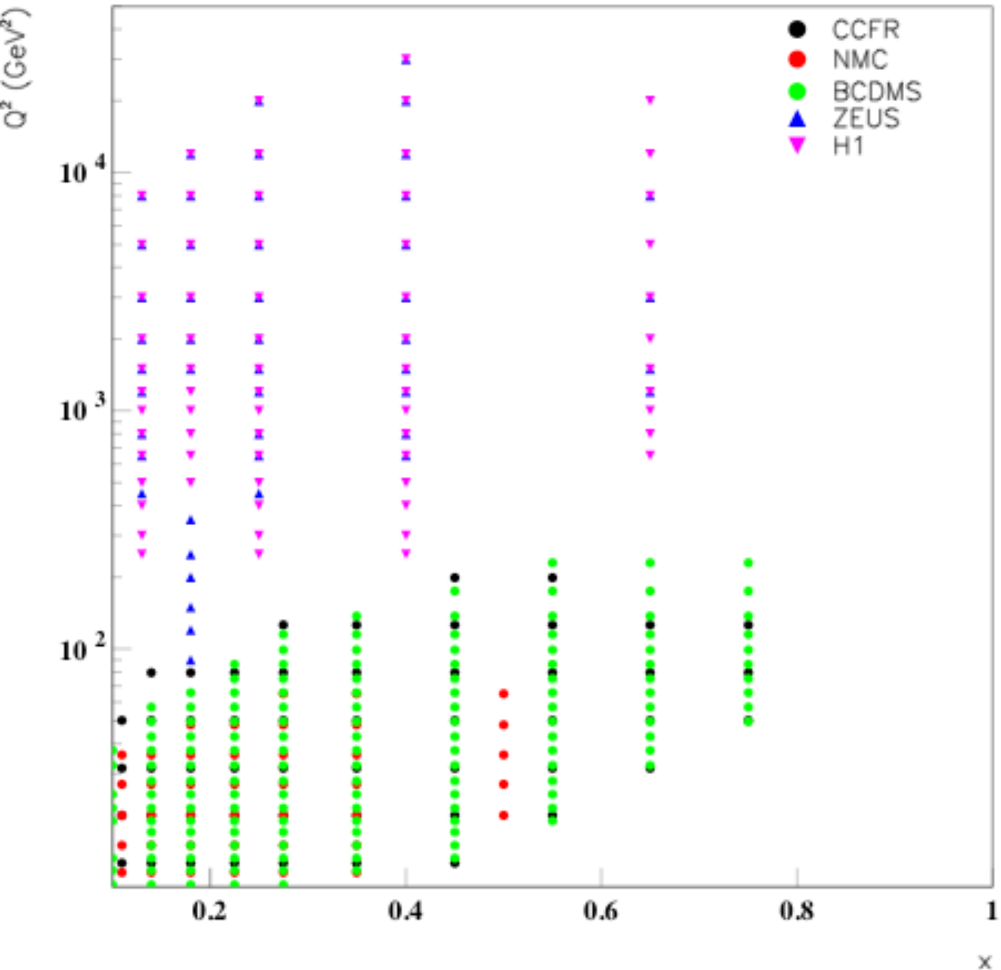
- ZEUS high-x data unique, but not used in PDF fits
- a transfer matrix formulation makes it possible to compare PDF set predictions to the ZEUS high-x data directly and calculate probabilities
- Forward modeling developed to extract information on PDFs from this data set
- First tests indicate that up-valence can be well constrained (momentum fraction and shape)
- Application to real data soon

QCDNUM code: https://github.com/cescalara/QCDNUM.jl
PDF fitting code: https://github.com/cescalara/PartonDensity.jl

BACKUP

Motivation

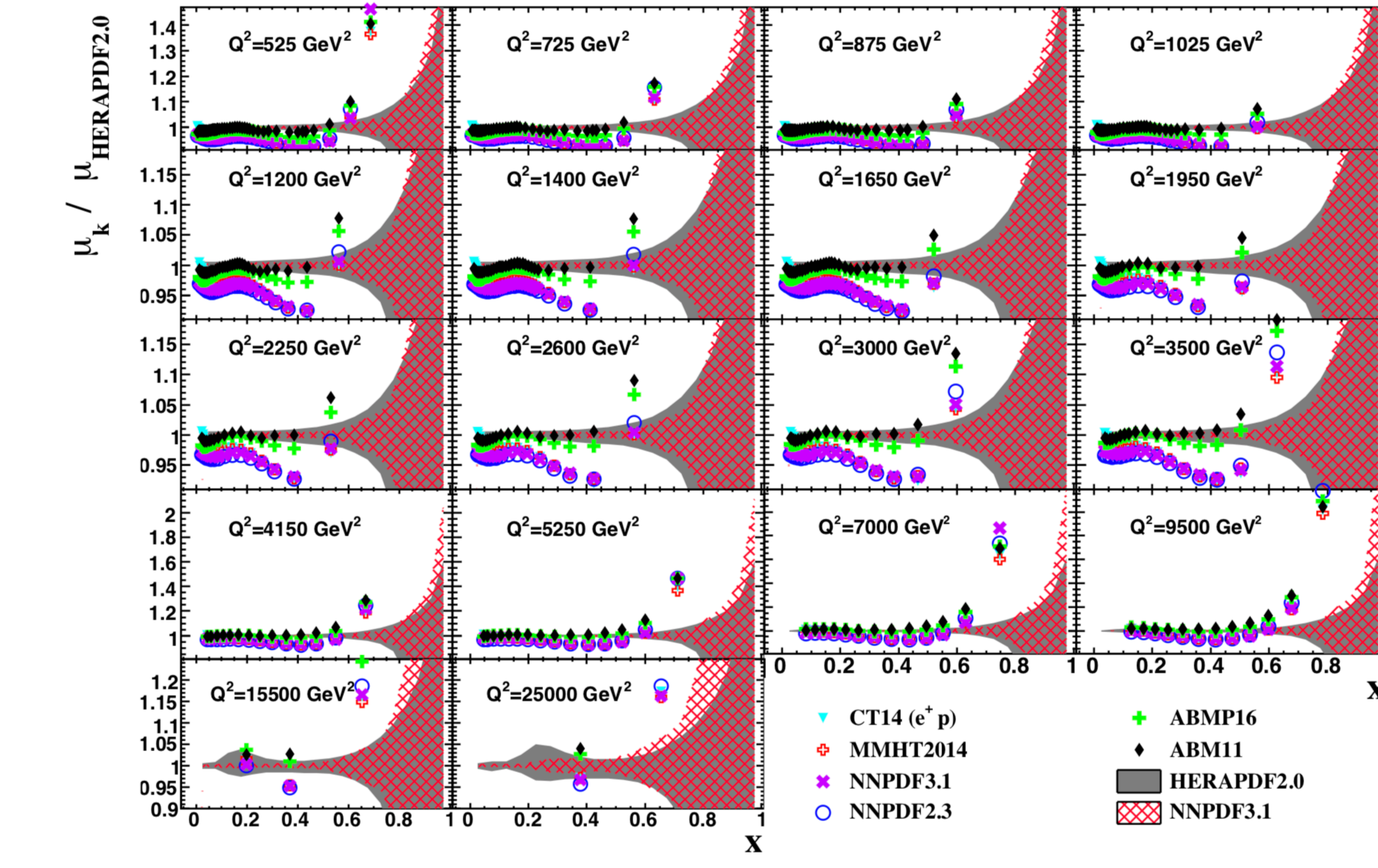
Information on the very high x behavior of the parton densities in the DGLAP validity regime is primarily theoretical and assumption-based.



BCDMS has measured F_2 up to x=0.75

The combined H1, ZEUS results are up to x=0.65

ZEUS has measured up to x=1, but these data are not (yet) included in PDF fits.



Sizable differences in expectations (much bigger than quoted uncertainties) despite the fact that fits typically use similar parametrization $xq \propto (1-x)^{\eta}$. Is it possible to improve this situation ?