Including factorisation and renormalisation scale -choice and -variation into FastNLO

The FastNLO idea

fastNLO

- FastNLO factorizes the cross section calculation for an a-posteriori inclusion of pdf's and alpha_s for jet-production
- The basic cross section formula for pQCD (DIS)

$$\sigma = \sum_{a,n} \int_0^1 dx \alpha_s^n(\mu_r) \cdot c_{a,n}(\frac{x_{Bj}}{x}, \mu_r, \mu_f) \cdot f_a(x, \mu_f)$$

  - n: order alpha_s (perturbative order)
  - a: number of flavors (internally it is Δ,g,Σ)
  - c: perturbative coefficients from the matrix elements
  - f: pdf

The FastNLO idea

fastNLO

- FastNLO factorizes the cross section calculation for an a-posteriori inclusion of pdf's and alpha_s for jet-production
- The basic cross section formula for pQCD (DIS)

$$\sigma = \sum_{a,n} \int_0^1 dx \alpha_s^n(\mu_r) \cdot c_{a,n}(\frac{x_{Bj}}{x}, \mu_r, \mu_f) \cdot f_a(x, \mu_f)$$

BUT: what about the scales ?

- the 'x'-integral is getting factorized

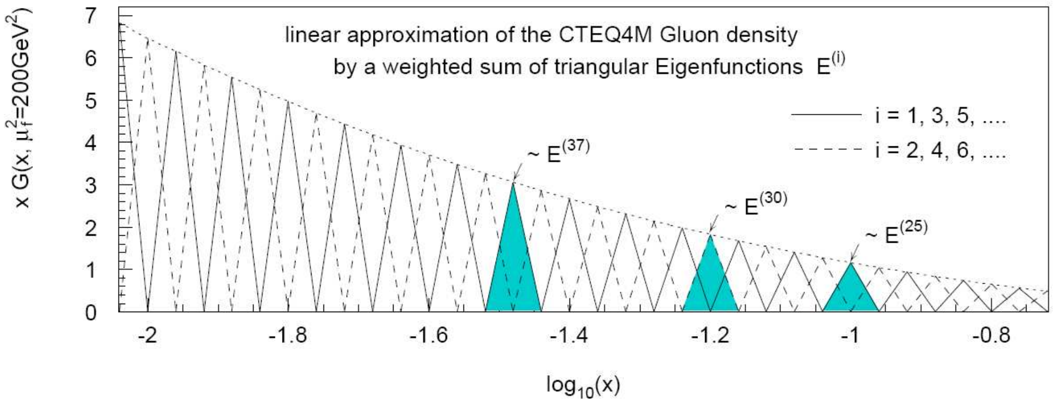
$$f_a(x) \cong \sum_i f_a(x_i) \cdot E^{(i)}(x)$$

- and calculated at some 'eigen'values

$$\sigma \cong \sum_{a,n,i} \alpha_s^n f_a(x_i) \underbrace{\int dx c_{a,n}(\frac{x_{Bj}}{x}) E^{(i)}(x)}_{\tilde{\sigma}} = \sum_{a,n,i} \alpha_s^n f_a(x_i) \tilde{\sigma}_{a,n}^{(i)}$$

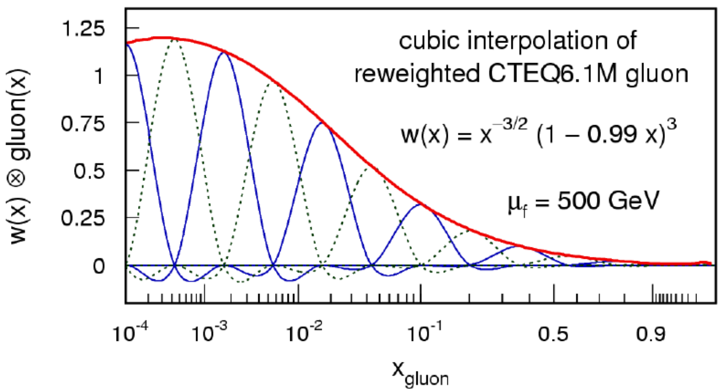
approximation of 'gluon' density

- as example: approximation by triangular eigenfunctions



$$E^{(i)}(x) = \begin{cases} 1 & x = x_i \\ \log(x_{i-1}) = \log(x) & x_{i-1} < x < x_i \\ \log(x_{i-1}) = \log(x) & x_i < x < x_{i+1} \\ 0 & x < x_{i-1}, \text{or } x > x_{i+1} \end{cases}$$

- FastNLO uses more sophisticated cubic eigenfunctions



- Basically: Each 'weight' is distributed over 4 scale nodes
- Afterwards, we only have to sum up those nodes

The scale dependence

- α_s and pdf have to be evaluated at the correct μ value

- FastNLO takes $\mu_r = \mu_f = \mu$
- FastNLO calculates cross section within 'sufficiently' small intervals of μ
  - those intervals are also interpolated using cubic eigenfunctions

$$\sigma = \sum_{a}^{flavors} \sum_{n}^{orders} \sum_{i}^{x-nodes} \sum_{j}^{\mu-nodes} \alpha_s^n(\mu(j)) \cdot \tilde{\sigma}_{a,n,j}^{(i)}(\mu(j)) \cdot f_a(x_i, \mu(j))$$

- Save another 'table' of μ-nodes μ(j)

- Those 'tables' are calculated for each bin of our measurement

Needs for improvements

- $\mu_r = \mu_f$ might not be wanted (e.g. low Q^2 jets)
- Some studies for different scales might be done e.g.
    - Q^2
    - P_T jet in Breit frame
    - <pT>all jets
    - sqrt [(Q^2+p_T2)/2]
      sqrt[(Q^2+p_T2)/4]
    - sqrt[(Q^2+p_T2)]
- Scales should be cosistent for combined fits
    - Zeus, H1, should not choose different scales in pdf fits
- Estimation of theory error
    - scales are varied by factor 2 up and down
    - BUT: sigma-tilde is scale dependent
    - Scale variations are only approximations

- Solution:

We store scale independent weights, and do all the scale calculations when calculating the cross section

cross section in nlojet++

- The cross section calculation in some detail

$$\sigma = \sigma^{born} + \sigma^{real} + \sigma^{virtual}(\mu_r, \mu_f)$$

$$\sigma^{virtual} = \sigma^{sub} + \sigma^{finix}(\mu_r, \mu_f) + \sigma^{fini\_1}(\mu_r, \mu_f)$$

- while the cross section is only a sum of weights in the MC integration

$$w = W_{PS} \cdot pdf \cdot \alpha_s \cdot \mathrm{M}(\mu_r, \mu_f) \cdot \alpha_{em}^2(Q^2) \cdot (\hbar c)^2$$

$$M^{finite} = M_0 + M_f \cdot \ln(\frac{\mu_f^2}{Q^2}) + M_r \cdot \ln(\frac{\mu_r^2}{Q^2})$$

- which is adapted in fastNLO like

$$w = W_{PS} \cdot {}^{1}\!/_{x} \cdot 1 \cdot M(\mu_r, \mu_f) \cdot \alpha_{em}^2(Q^2) \cdot (\hbar c)^2 \equiv c \cdot M$$

improved FastNLO

- Access M_0, M_f, and M_r directly
  - slight hack of nlojet++
- Store THREE tables for
  - M_0*c -> σ()
  - M_f *c-> $\sigma_f(\mu_f)$
  - M_r *c-> $\sigma_r(\mu_r)$
  - do the multiplication the same way as nlojet++ does it

- Calculating the cross section

$$\sigma = \sigma_0 + \sigma_f \cdot \ln(\frac{\mu_f^2}{Q^2}) + \sigma_r \cdot \ln(\frac{\mu_r^2}{Q^2})$$

  - remember: σ_{xy} here is actually a sum over x, n, a, (i)

- Need for knowledge of Q^2, μ_r and μ_f
  - -> Additional "Scale" table for Q^2
- Better
  - Additional "Scale" table for Q AND one table for p_T !!!

new final formula

- FastNLO formula with three scale independent tables σ_0, σ_f, σ_r and two 'scale' lookup tables (q,p)

$$\sigma_{Bin} = \sum_{a}^{flavors} \sum_{n}^{orders} \sum_{i}^{x-nodes} \sum_{q}^{Q-nodes} \sum_{p}^{p_T-nodes} \begin{pmatrix} \alpha_s^n(\mu_r(q,p)) \cdot \sigma_{0\,a,n}^{(i)} \cdot f_a(x_i, \mu_f(q,p)) + \\ \alpha_s^n(\mu_r(q,p)) \cdot \sigma_{f\,a,n}^{(i)} \cdot \ln(\frac{\mu_f(q,p)^2}{Q(q)^2}) \cdot f_a(x_i, \mu_f(q,p)) + \\ \alpha_s^n(\mu_r(q,p)) \cdot \sigma_{r\,a,n}^{(i)} \cdot \ln(\frac{\mu_r(q,p)^2}{Q(q)^2}) \cdot f_a(x_i, \mu_f(q,p)) \end{pmatrix}$$

  - while μ_r and μ_f are just some functions of the stored (in the 'scale table') Q and p_T values at p and q
    - e.g. μ_r (q,p) = sqrt((Q_q^2+$p_{T,p}^2$)/2)
  - one could also store sth. different than p_T but one always needs Q^2

pp and ppbar

- somehow simpler and more complicated
  - sum over x_1 and x_2 for both hadrons
  - 7(6) types of pdf – linear combinations
  - 6 types of contributions to xs formula
    - born, real, sub, finix1, finix2, fini1
- μ_r and μ_f dependence is more trivial

$$\sigma = \sigma_0 + \sigma_f \cdot \log(\mu_f^2) + \sigma_r \cdot \log(\mu_r^2)$$

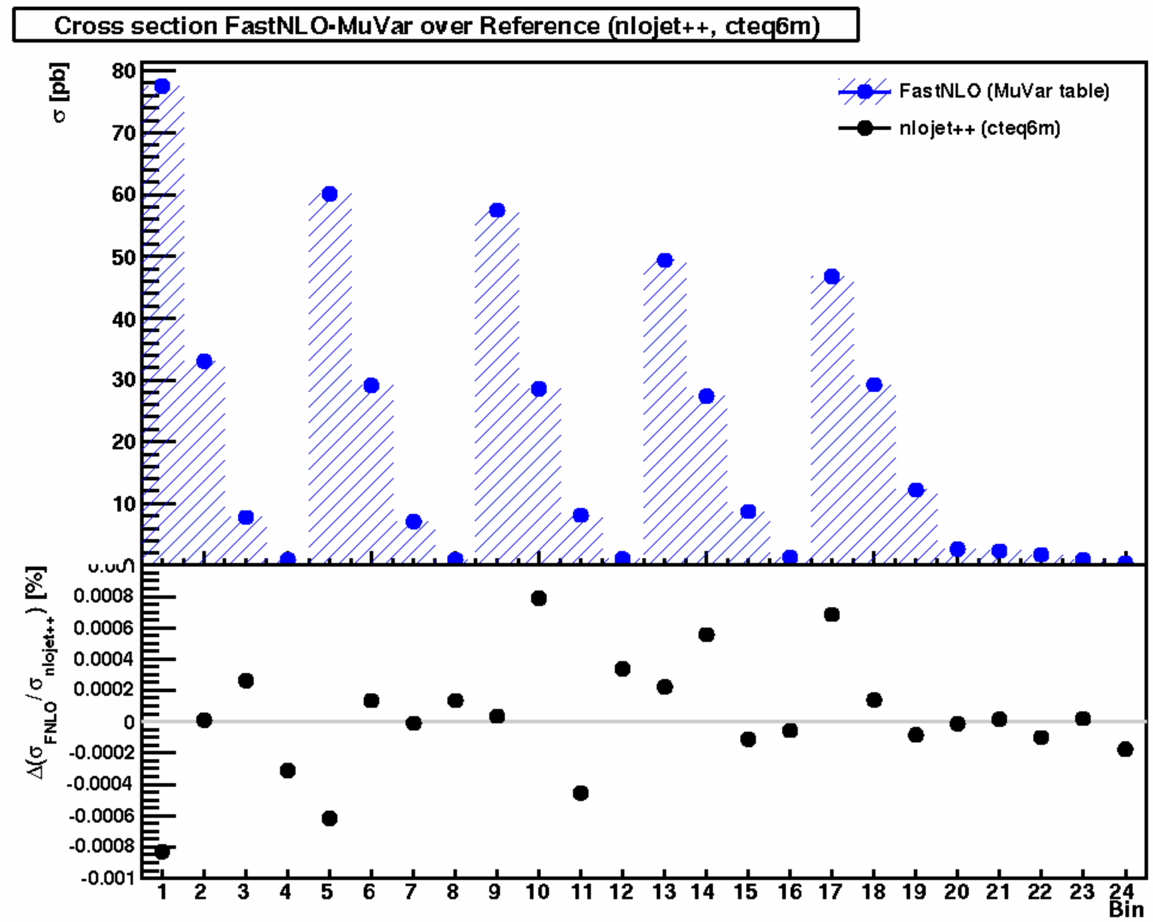
- final formula is rather simple then

Make a long story short

- Testing with inclusive jets at high Q^2
- choosing
  - 70 x-nodes
  - 30 nodes for Q
  - 30 nodes for p_T
- Using always cubic interpolation
  - each weight is distrubuted over 4x4x4 nodes

- choosing (!!) $\mu_r = \mu_f = Q^2$
- Tablesize: 200 MB

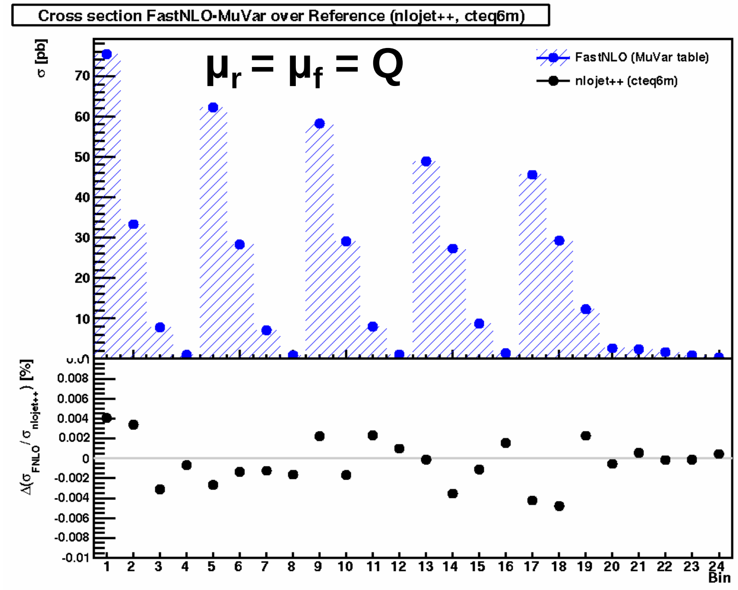
- **FNLO precision**
  - **better than: 0.0005%**



Use less bins, and study scale choice

- choosing
  - 30 x-nodes
  - 20 nodes for Q
  - 15 nodes for pT
  - size: 30MB
- use the appropriate reference cross section

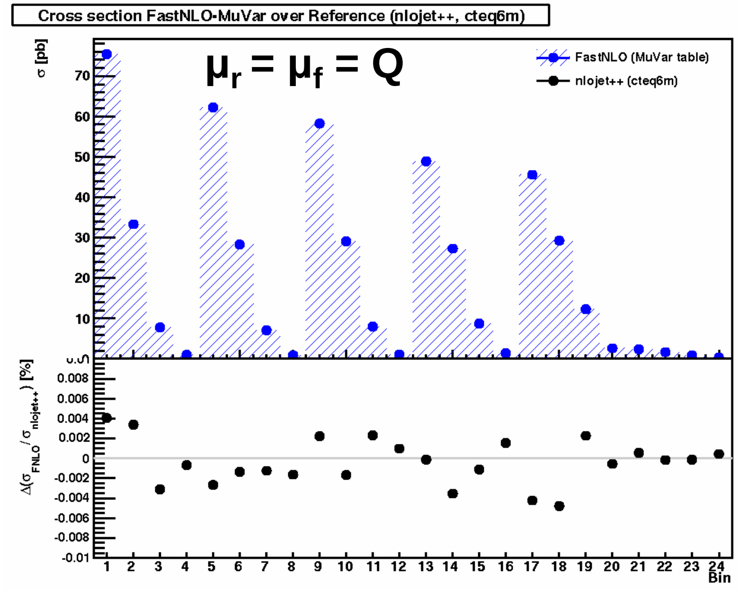
- Precision is mostly dependent on number of x-bins:
  - "High Q2, High pT" bins span over smaller x-range -> higher precision with fixed number of bins (Backup)
  - Different number of x-nodes per bin is already implemented!

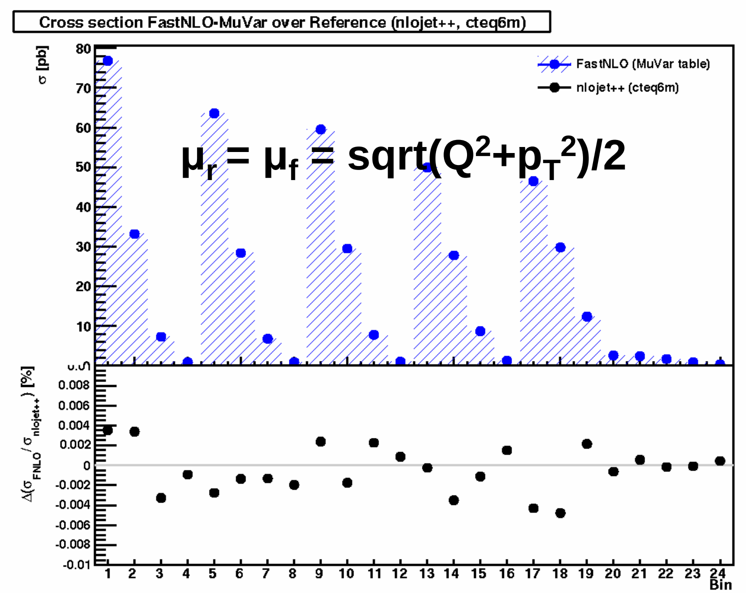


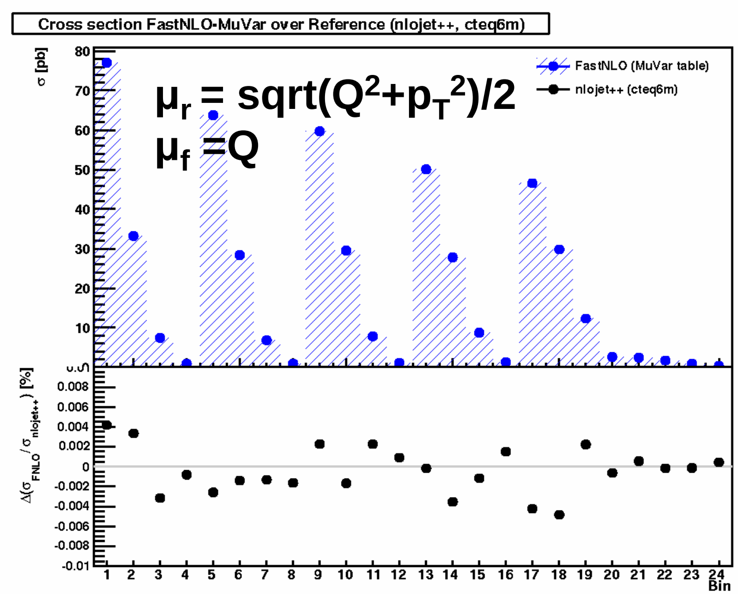
Use less bins, and study scale choice

- choosing
  - 30 x-nodes
  - 20 nodes for Q
  - 15 nodes for pT
  - size: 30MB
- use the appropriate reference cross section

- Precision is mostly dependent on number of x-bins:
  - "High Q2, High pT" bins span over smaller x-range -> higher precision with fixed number of bins (Backup)
  - Different number of x-nodes per bin is already implemented!

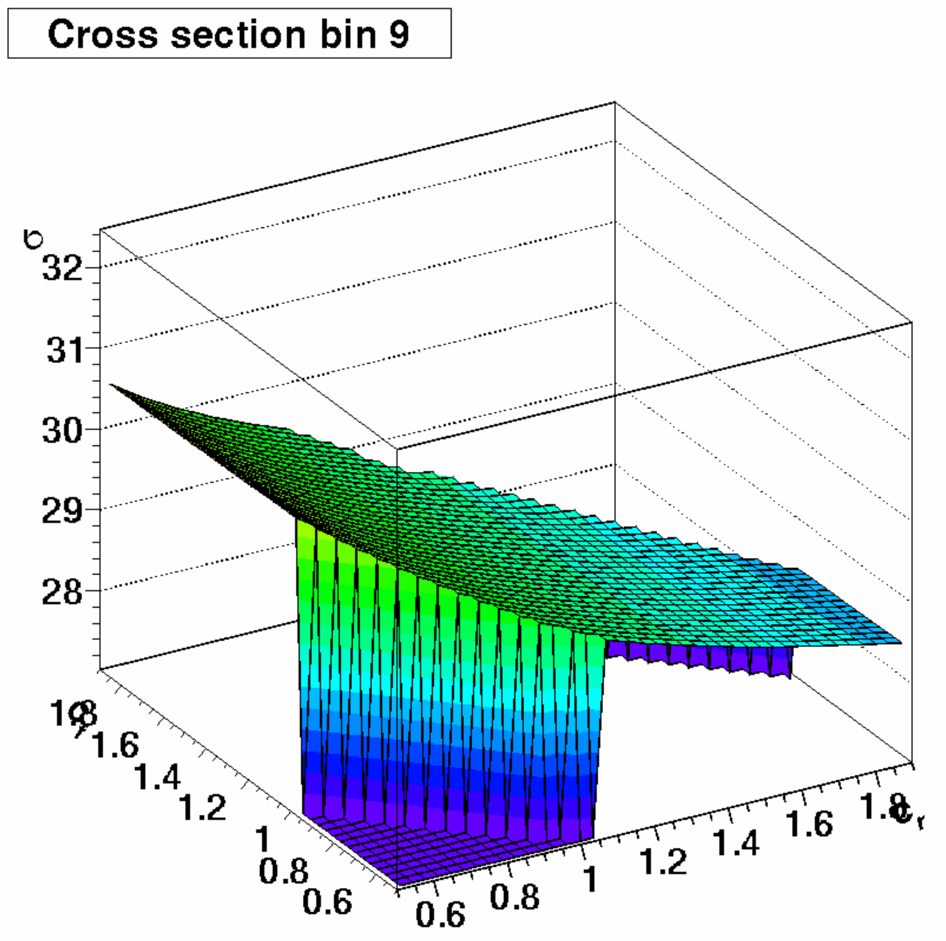






Scale variation for 'theory error'

- functional form of scales is free to choose
  - also $c_r*\mu_r$ and $c_f*\mu_f$ is a valid choice
- studying scale variation with very high precision and very high statistics!
  - constrain: $0.5 < c_r*c_f < 2.0$
- new and very precise determination of conventionally defined 'theory error'



Comparision

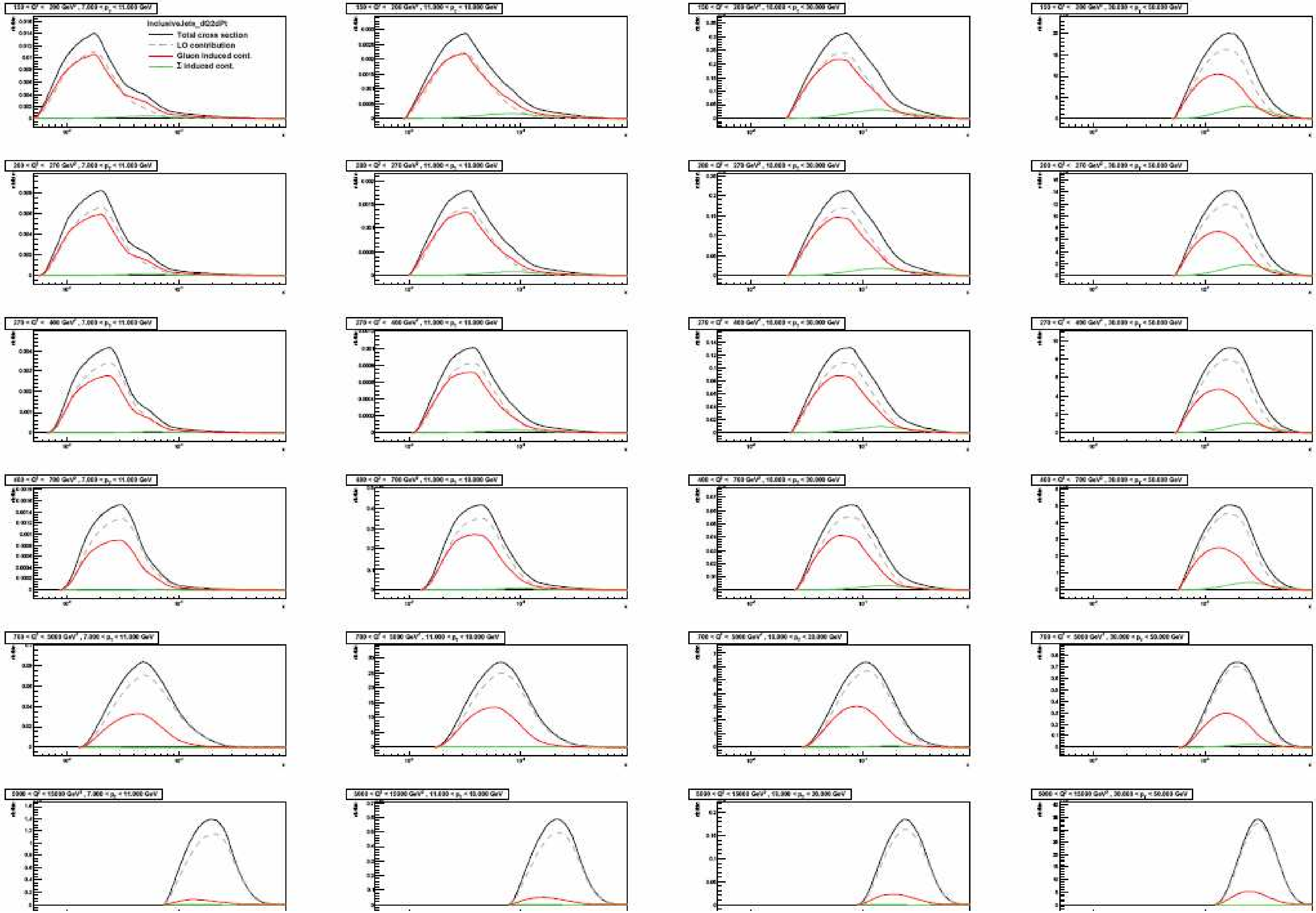
	FastNLO (to be v2.0)	This FastNLO mod
Size of table (speed)	$n_{Bins} \times n_x \times n_{mur}$ [$\times n_{muf}$]$\times$ $n_{scalevar} \times n_{proc} \times n_{ord}$	$n_{Bins} \times n_x \times n_{Pt} \times n_{Q2} \times n_{proc} \times n_{ord} \times 3$
choice of renormalization scale	Fixed when creating table e.g. $\mu_r^2=(Q^2+p_T^2)/2$	Any function using Q^2 and Pt is possible (or any other variables used, when creating table)
choice of factorization scale	equals renormalization scale (v2.0) (I have a mod, where $\mu_r \neq \mu_f$)	Any function using Q^2 and Pt is possible (or any other variables used, when creating table)
variation of μ_r (e.g. ×2, ×0.5)	fixed, when creating table	every factor is possible
variation of μ_f (e.g. ×2, ×0.5)	fixed and must be equal μ_r	every factor is possible (independent of μ_r)
free variation of scale factor	μ_r: only in an approximation μ_f: not supported	see above
How to determine theory uncertainty	vary μ_r and μ_f up and down simultaneously by ×2 and ×0.5	vary μ_r and μ_f independently with $0.5<c_r*c_f<2.0$ and look for highest and lowest prediction (e.g. ATLAS jets: arXiv:1009.5908)

Summary

- The FastNLO concept was improved
- Scale independent tables allow to choose renormalization and factorization scale afterwards almost arbitrary
  - a third variable (like Θ, pt-leading) could be implemented
- Scale variations for theory error determination can be performed much more sophisticated and without any approximation
- The precision of this FastNLO version can reach more than 0.0005%
- This method is implemented and well tested for DIS processes
- It is already implemented for pp and ppbar but testing is needed (reader is missing)

- Authors of FastNLO did not answer my last eMail about that
- I would propose to use this method for our upcoming alpha_s fits
- Method is also great for combined fits and cross checks with ZEUS, if they use different scales than we do

Backup


InclusiveJets_dQ2dPt
Total cross section
LO contribution
Gluon induced cont.
Σ induced cont.