Universität Münster

PineAPPL Grids of Open Heavy-Flavor Production in the GM-VFNS

Deep-Inelastic Scattering 2024

Jan Wissmann

nCTEQ nuclear parton distribution functions

Graduiertenkolleg 2149 Research Training Group

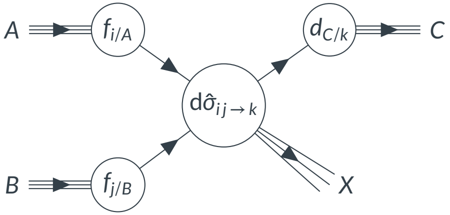
9th of April, 2024

Introduction

- Process: Open heavy-quark hadroproduction

$$A + B \to C + X$$

Hadrons (p, Pb, ...) → A, B; Heavy meson (D, B, ...) → C



in collinear factorization:

$$\mathrm{d}\sigma = f_{i/A} \otimes f_{j/B} \otimes \mathrm{d}\hat{\sigma}_{ij\to k} \otimes d_{C/k}$$

- Light quarks q: u, d, s, heavy quarks Q: c, b
  → heavy on the absolute QCD scale: $m_\mathrm{Q} \gg \Lambda_\mathrm{QCD}$ so that the process is calculable perturbatively, i.e. $\alpha_s(m_\mathrm{Q}) \ll 1$
- Importance of heavy-quark production: data goes to small momentum-fraction $x \approx \frac{p_\mathrm{T}}{\sqrt{s}} e^y \sim 10^{-5}$
  → e.g. constrain gluon PDF in low-x region
- Mass effects non-negligible for $p_\mathrm{T} \sim m_\mathrm{Q}$
- Theory predictions for this process: GM-VFNS (NLO)

Flavor-number schemes

- In all processes with heavy quarks: new scale m_Q

threshold region — $p_T \sim m_Q$ → asymptotic region — $p_T \gg m_Q$ → p_T

FFNS
Fixed flavor-number scheme

- Heavy quark treated as massive particle, lighter quarks as massless partons
- Fixed number of light flavors
- Non-zero mass acts as regulator

ZM-VFNS
Zero-mass variable-flavor-number scheme

- Heavy quarks treated as massless partons
- Number of light flavors is scale-dependent: Contributions from new flavors activate dynamically at their respective mass thresholds
- Collinear singularities due to massless quarks renormalized in the usual $\overline{\text{MS}}$

The General-Mass Variable-Flavor-Number Scheme (GM-VFNS)

Expectation:

$$\mathrm{d}\sigma_{\mathrm{FFNS}} \xrightarrow{p_T \gg m_Q} \mathrm{d}\sigma_{\mathrm{ZM-VFNS}} \quad ?$$

→ $p_T \gg m_Q$ (i.e. $m_Q \to 0$) limit and subtraction of collinear singularities are not exchangeable

Solution: GM-VFNS:

$$\mathrm{d}\sigma_{\mathrm{FFNS}} \xleftarrow{m_Q \leftarrow p_T} \mathrm{d}\sigma_{\mathrm{GM-VFNS}} \xrightarrow{p_T \gg m_Q} \mathrm{d}\sigma_{\mathrm{ZM-VFNS}}$$

For intermediate p_T, the GM-VFNS interpolates between the ZM-VFNS and the FFNS.

Gridding with PineAPPL

For theoretical predictions obtained with a Monte Carlo (MC) generator:

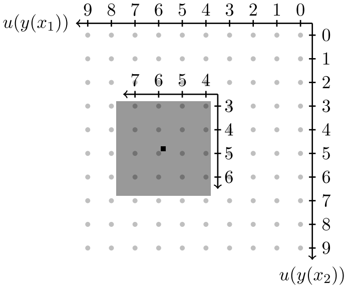
Problem:

- A-posteriori variation of α_s, scales and PDFs requires running the MC generator again each time (usually multiple hours per run)
- Same calculation of the hard-scattering matrix elements is performed every time

Solution:

- Pre-calculate the MC weights and store them in an interpolation "grid" independent of the PDFs, α_s and possibly scales

→ Done by libraries such as
  - APPLgrid [0911.2985]
  - FastNLO [hep-ph/0609285]
  - PineAPPL [2009.03987]



[2009.03987]

Gridding with PineAPPL II

Gridding libraries store the MC weights of fixed-order calculations by interpolating the PDFs:

$$f(x) = \sum_i f_i \, L_i(x) \qquad \text{(analogous for } \alpha_s \text{ and scales)}$$

↑ Lagrange interpolation basis functions

1. Gridding Stage

- Build grid by integrating over the basis functions:

$$d\sigma_i = \int dx \, L_i(x) \, d\hat{\sigma}(x)$$

- In practice: Fill the grid with the MC weights, obtain grid file

→ Takes multiple CPU hours, but only done once

2. Convolution Stage

- Obtain predictions by performing the convolution with the PDF:

$$d\sigma = \sum_i f_i \, d\sigma_i$$

→ ~Instantaneous, can be done multiple times for different PDFs and scales

Gridding GM-VFNS heavy-quark hadroproduction

- The NLO GM-VFNS calculation in heavy-quark hadroproduction exists as Fortran code by B. A. Kniehl, G. Kramer, I. Schienbein, H. Spiesberger [hep-ph/0502194] [hep-ph/0502194]
- Our work: Extending the existing code to produce PineAPPL grids and writing a Python interface to the code to make it publication-ready

NEW: produced NLO GM-VFNS predictions as PineAPPL grids

One grid, corresponding to one experimental dataset, includes cross-sections...

- double-differential in (p_T, y) corresponding to the bins of the experimental data
- at LO (α_s^2) and NLO (α_s^3)
- with the FF baked-in (Since PineAPPL allows up to two different convolutions at the moment)

→ both PDFs and α_s can be varied a-posteriori, e.g. for PDF uncertainties or fits

PLANNED:

- Using these grids for nCTEQ PDF analyses in the future
- Publication of the grids and this version of the GM-VFNS code

Data taken into account so far – ALICE

Predictions and grids already produced for:

Experiment	arXiv	Initial State	Meson
ALICE	1111.1553	p + p	D^0
	1405.3452	p + Pb	D^0
			D^+
	1605.07569	p + p	D^0
		p + Pb	D^+
	1702.00766	p + Pb	D^0
	1901.07979	p + p	D^0
	1906.03425	p + Pb	D^+
			D_s^+
			D^{*+}
	2106.08278	p + p	D^0

Fragmentation functions:

D^0, D^+, D^{*+}:	KKKS08	[0712.0481]
D_s^+:	BKK06_D	[hep-ph/0607306]
B^+:	BKK06_B	[0712.0481]

Data taken into account so far – LHCb & CMS

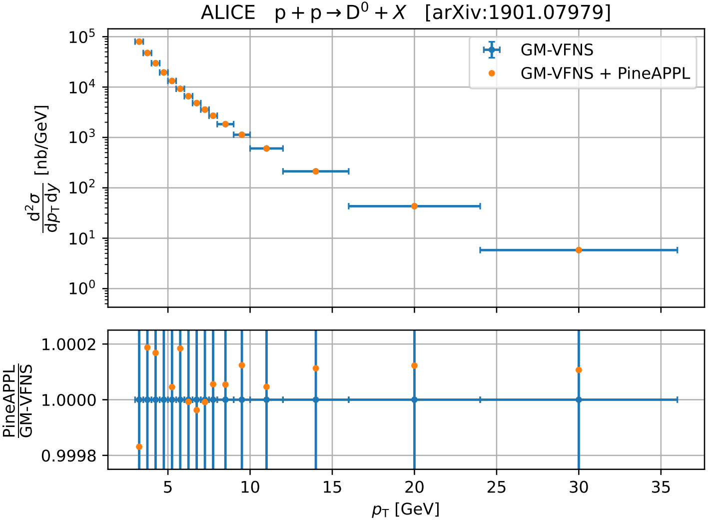
Predictions and grids already produced for:

Experiment	arXiv	Initial State	Meson
CMS	1508.06678	p + Pb	B^+
LHCb	1302.2864	p + p	D^0
	1510.01707	p + p	D^0
	1610.02230	p + p	D^0
	1707.02750	p + Pb	D^0
	2205.03936	p + Pb	D^0

Fragmentation functions:

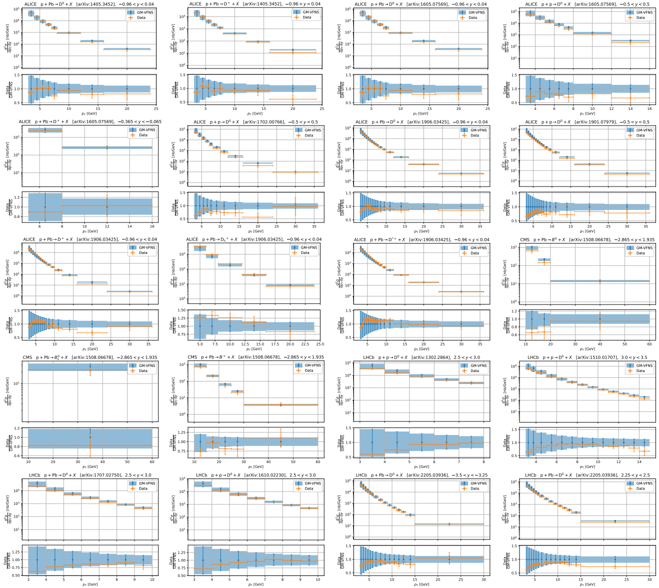
D^0, D^+, D^{*+}:	KKKS08	[0712.0481]
D_s^+:	BKK06_D	[hep-ph/0607306]
B^+:	BKK06_B	[0712.0481]

Results I – Prediction vs. Grid



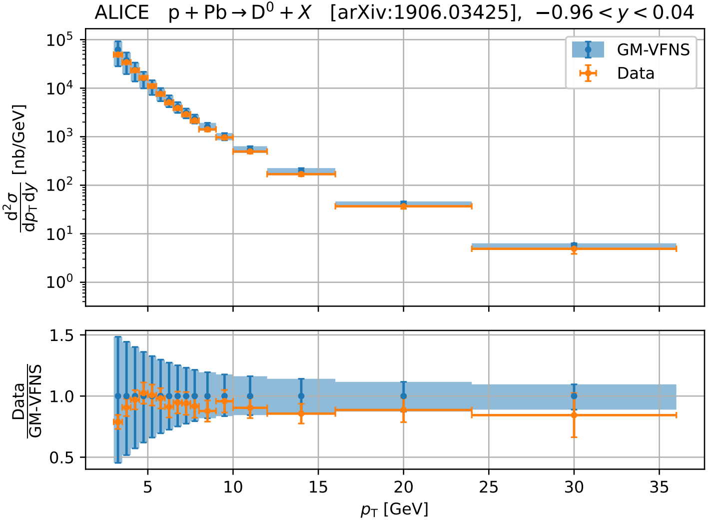
- Sub-permille agreement
- Shown here: Statistical (MC) errors
- Grid precision independent of run settings and phase space region

Results II – Prediction vs. Data



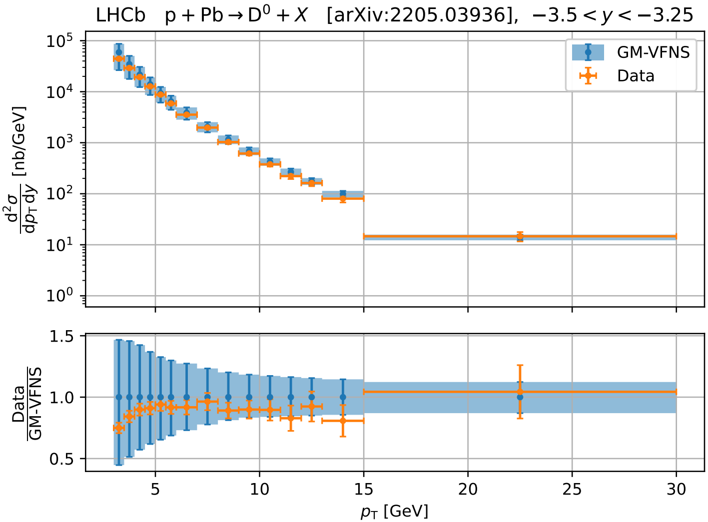
- Next slides: Some examples of the predictions

Results II – Prediction vs. Data



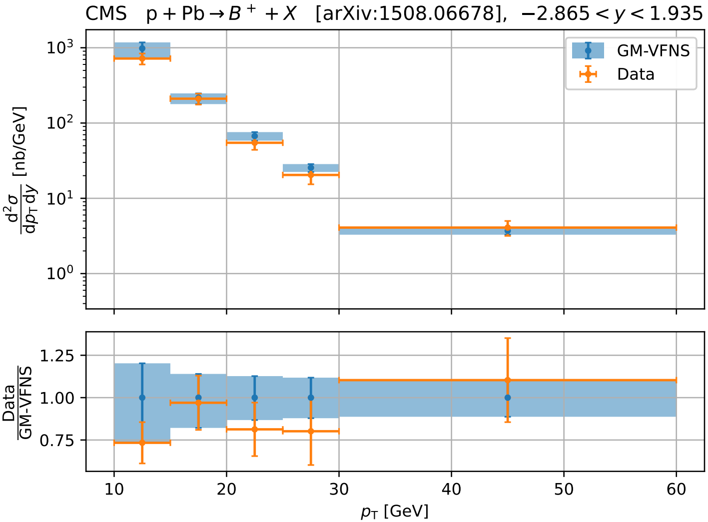
- 7-point (ξ_r, ξ_f) scale-variation where $\mu_i = \xi_i \sqrt{p_T^2 + 4m_Q^2}$ and $\xi_i \in \{0.5, 1, 2\}$
- Previous work by the GM-VFNS authors: Scale choice improves agreement and improves predictions at lower p_T

Results II – Prediction vs. Data



- 7-point (ξ_r, ξ_f) scale-variation where $\mu_i = \xi_i \sqrt{p_T^2 + 4m_Q^2}$ and $\xi_i \in \{0.5, 1, 2\}$
- Previous work by the GM-VFNS authors: Scale choice improves agreement and enables meaningful predictions at lower p_T

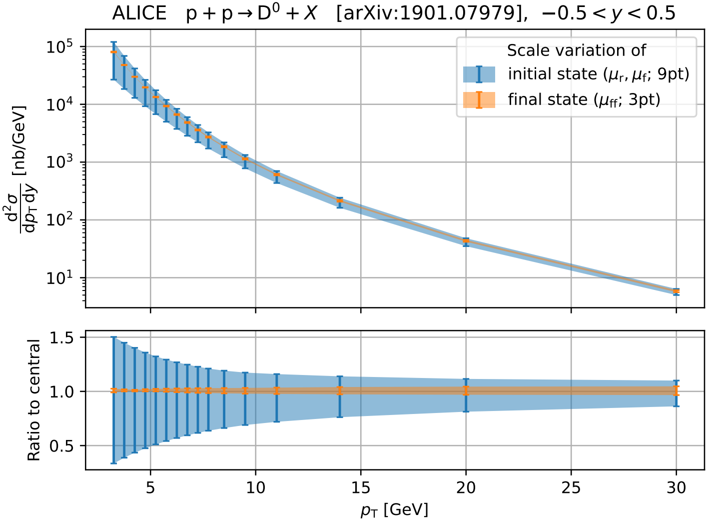
Results II – Prediction vs. Data



- 7-point (ξ_r, ξ_f) scale-variation where $\mu_i = \xi_i \sqrt{p_T^2 + 4m_Q^2}$ and $\xi_i \in \{0.5, 1, 2\}$
- Previous work by the GM-VFNS authors: Scale choice improves agreement and enables meaningful predictions at lower p_T

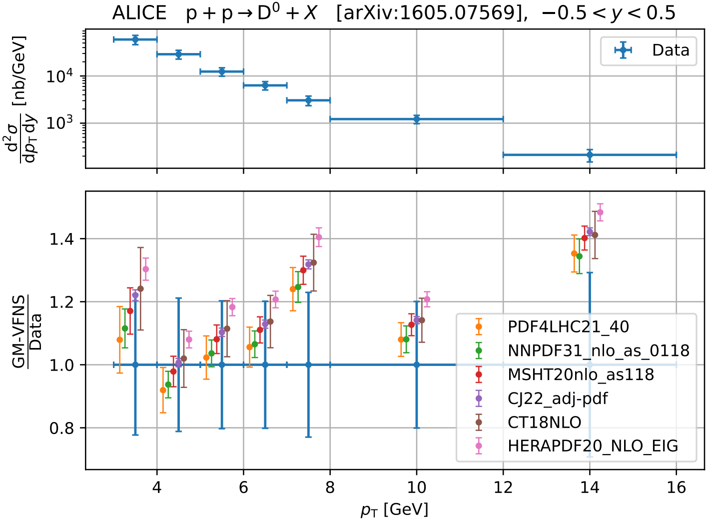
Results III – Fragmentation Scale

$$d\sigma = f_{i/A}(\mu_f) \otimes f_{j/B}(\mu_f) \otimes d\hat{\sigma}_{ij\to k}(\mu_r) \otimes d_{C/k}(\mu_{ff})$$



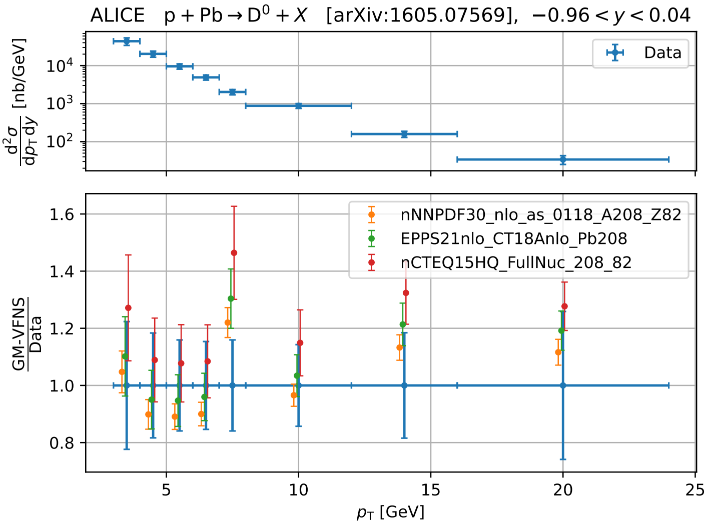
- 7-point (ξ_r, ξ_f) scale-variation where $\mu_i = \xi_i \sqrt{p_T^2 + 4m_Q^2}$ and $\xi_i \in \{0.5, 1, 2\}$
- Previous work by the GM-VFNS authors: Tuned scales improve agreement and enable meaningful predictions at lower p_T

Results IV – PDF Uncertainties



- Total number of PDF members: 342
- Here: Gridding reduces execution time by factor 342 → impossible without gridding

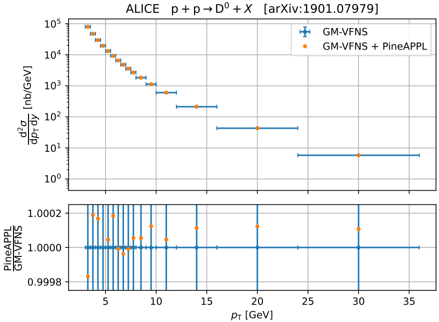
Results IV – PDF Uncertainties



- Total number of PDF members: 347
- Here: Gridding reduces execution time by factor 347 → impossible without gridding

Conclusion

- The GM-VFNS gives the heavy-quark production prediction for a bigger kinematic range
- Gridding libraries like PineAPPL allow varying the PDFs and scales a-posteriori in a very efficient way
- The GM-VFNS prediction and the produced grids agree by less than one per-mille
- This version of the GM-VFNS code and the PineAPPL grids will be published



Backup

Grid interpolation – Details

- The MC generator generates events with weights W_m corresponding to points (x_{1m}, x_{2m}, Q_m^2):

$$\begin{aligned} d\sigma &= \sum_{i,j} \int dx_1\, dx_2\, dQ^2\, f_i(x_1)\, f_j(x_2)\, d\hat{\sigma}_{ij} \\ &= \sum_{i,j,m,n} f_i(x_{1m}, Q_m^2)\, f_j(x_{2m}, Q_m^2)\, \alpha_s^{p_n}(Q_m^2)\, W_{m,ij}^{(n)} \end{aligned} \tag{1}$$

- Precalculating the weights would let us perform the convolution at a later stage

→ We can store the weights W_m for each histogram bin in a lookup table (grid) together with the points (x_{1m}, x_{2m}, Q_m^2)

- To make this space efficient, we Lagrange-interpolate between grid points in a transformed (x_1, x_2, Q^2)-space