

| **Yadism**
Yet Another DIS Module

New Software Tools for DGLAP and DIS

Alessandro Candido, Felix Hekhorn, Giacomo Magni

DIS 2022, Santiago di Compostela, May 2022

Acknowledgement: This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement number 740006.

Outline

1. EKO [arXiv: 2202.02338]
2. Intrinsic Charm in the Proton [submitted]
3. yadism [in preparation]
4. Theory Prediction Pipeline

1. EKO [arXiv: 2202.02338]



DGLAP:

$$\mu_F^2 \frac{d\mathbf{f}}{d\mu_F^2}(\mu_F^2) = \mathbf{P}(a_s(\mu_R^2), \mu_F^2) \otimes \mathbf{f}(\mu_F^2)$$

as operator equation for the evolution kernel operator (EKO) \mathbf{E} :

$$\mu_F^2 \frac{d}{d\mu_F^2} \mathbf{E}(\mu_F^2 \leftarrow \mu_{F,0}^2) = \mathbf{P}(a_s(\mu_R^2), \mu_F^2) \otimes \mathbf{E}(\mu_F^2 \leftarrow \mu_{F,0}^2)$$

with

$$\mathbf{f}(\mu_F^2) = \mathbf{E}(\mu_F^2 \leftarrow \mu_{F,0}^2) \otimes \mathbf{f}(\mu_{F,0}^2)$$

EKO Physics Features

- independent of boundary condition → PDF fitting
- Mellin (N -) space solution, but momentum (x -) space delivery via piecewise Lagrange-interpolation
- Intrinsic heavy quark distributions → see part 2
- Backward VFNS evolution (i.e. across thresholds and with intrinsic)
→ see part 2

EKO Project Management

The image shows two side-by-side screenshots. On the left is the GitHub repository page for 'EKO Evolution Kernel Operators'. It displays a list of pull requests, issues, and discussions. A prominent pull request from 'ghoul' is shown, titled 'Debug evolution: remove debug behavior', with a detailed commit message. Below the pull request list is the 'README.md' file, which contains the EKO logo and a brief description: 'EKO is a Python module to solve the DOLFIN equations in N-space in terms of Evolution Kernel Operators'.

On the right is the 'Interpolation' section of the EKO documentation. The title 'Interpolation' is at the top, followed by a sub-section 'Implementation: via interpolation'. It explains that to obtain the operators in an PDF-independent way, we use approximation theory. Below this is a code snippet showing the definition of the interpolation grid:

```

$$\Omega := \{x_j : 0 < x_j <= 1, j = 0, \dots, N_{\text{grid}} - 1\}$$

```

Further down, it defines the basis functions:

$$f(x) \approx \tilde{f}(x) = \sum_{j=0}^{N_{\text{grid}}-1} f(x_j)p_j(x)$$

It notes that each grid point x_j has an associated interpolation polynomial $p_j(x)$ (represented by `via_interpolation`). It then describes the 'Lagrange Interpolation' method, mentioning conditions on 'Crossing The Grid' and 'References'.

The 'Algorithm' section follows, stating that first, the interpolation region is split into several areas (represented by `via_interpolation_area`), which are located by the grid points:

$$A_j = [x_j, x_{j+1}], \quad \text{for } j = 0, \dots, N_{\text{grid}} - 2$$

It notes that while we include the right border point into the definition, but not the left which keeps all areas disjoint. This assumption is based on the physical fact that PDFs do have a fixed upper bound ($|x| < 1$), but no lower bound.

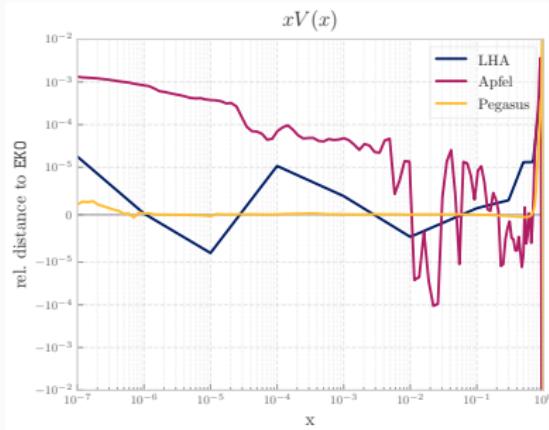
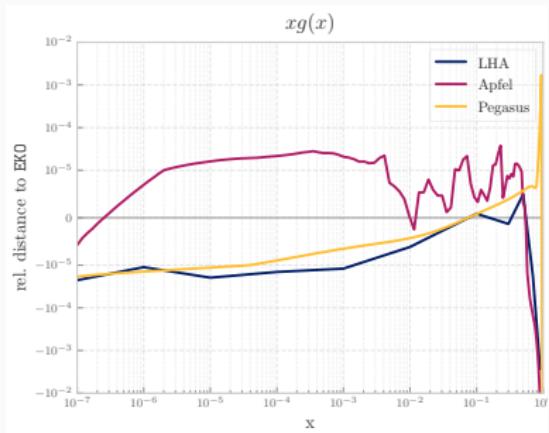
Finally, it describes the interpolation labels, which will build the interpolation polynomials and estimate the needed amount of points:

$$S_j = \{x_j, x_{j+1}\}, \quad \text{for } j = 0, \dots, N_{\text{grid}} - 2$$

- Fully open source: <https://github.com/N3PDF/eko>
- Written in Python
- Fully documented: <https://eko.readthedocs.io/>
- A cornerstone in a new theory prediction suite → see part 4

EKO Benchmarks

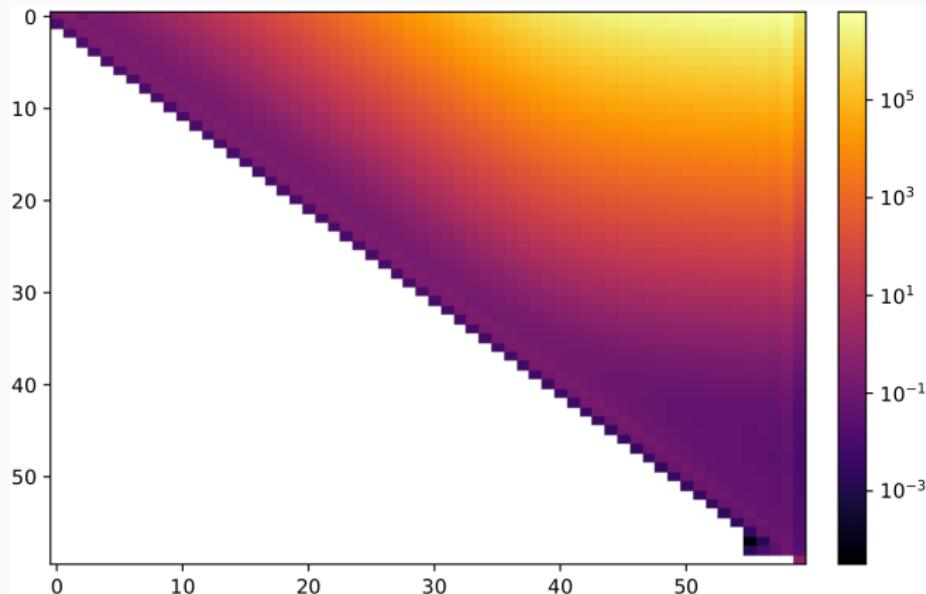
LHA benchmark [G+02][D+05]:



⇒ EKO is working!

EKO Snapshot: $\Sigma \leftarrow \Sigma$

FFNS LO LHA settings: $\Sigma(Q^2 = 10^4 \text{ GeV}^2) \leftarrow \Sigma(Q^2 = 2 \text{ GeV}^2)$



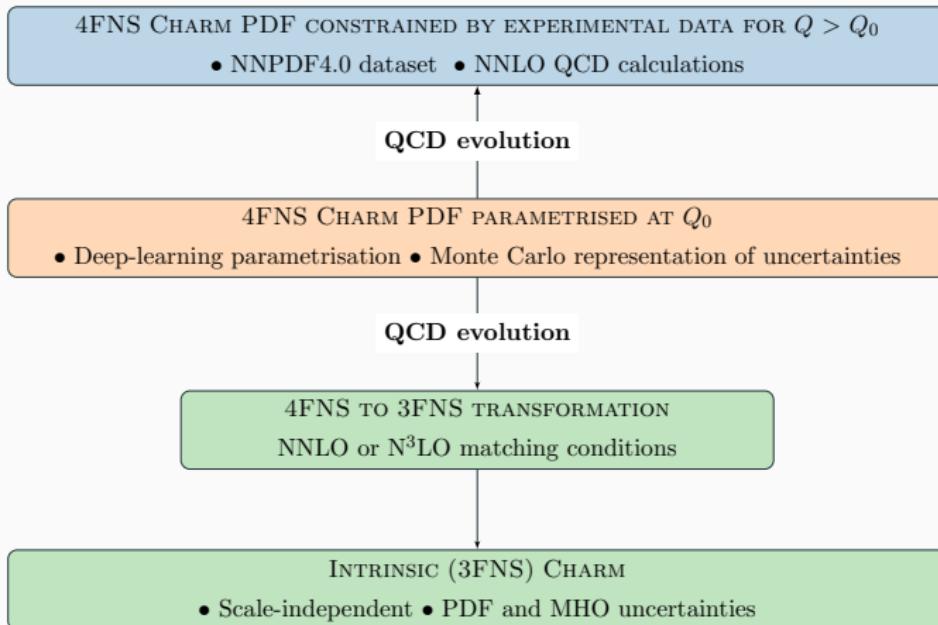
xgrid: $10^{-7}(0) \rightarrow \log \rightarrow \text{lin} \rightarrow 1(59)$,
axis: $x \sim \text{input } (2 \text{ GeV}^2), y \sim \text{output } (10^4 \text{ GeV}^2)$

2. Intrinsic Charm in the Proton

[submitted]

Intrinsic Charm: Strategy

- see poster by G. Magni
- based on NNPDF4.0 [[arxiv:2109.02653](https://arxiv.org/abs/2109.02653)] - see talk by R. Stegeman



Matching Conditions and Backward Evolution

For (forward) evolution across a matching scale μ_h^2 :

$$\tilde{\mathbf{f}}^{(n_f+1)}(\mu_{F,1}^2) = \tilde{\mathbf{E}}^{(n_f+1)}(\mu_{F,1}^2 \leftarrow \mu_h^2) \mathbf{R}^{(n_f)} \tilde{\mathbf{A}}^{(n_f)}(\mu_h^2) \tilde{\mathbf{E}}^{(n_f)}(\mu_h^2 \leftarrow \mu_{F,0}^2) \\ \times \tilde{\mathbf{f}}^{(n_f)}(\mu_{F,0}^2)$$

with $\mathbf{R}^{(n_f)}$ a flavor rotation matrix and $\tilde{\mathbf{A}}^{(n_f)}(\mu_h^2)$ the operator matrix elements (partially known up to N³LO)

Matching Conditions and Backward Evolution

For (forward) evolution across a matching scale μ_h^2 :

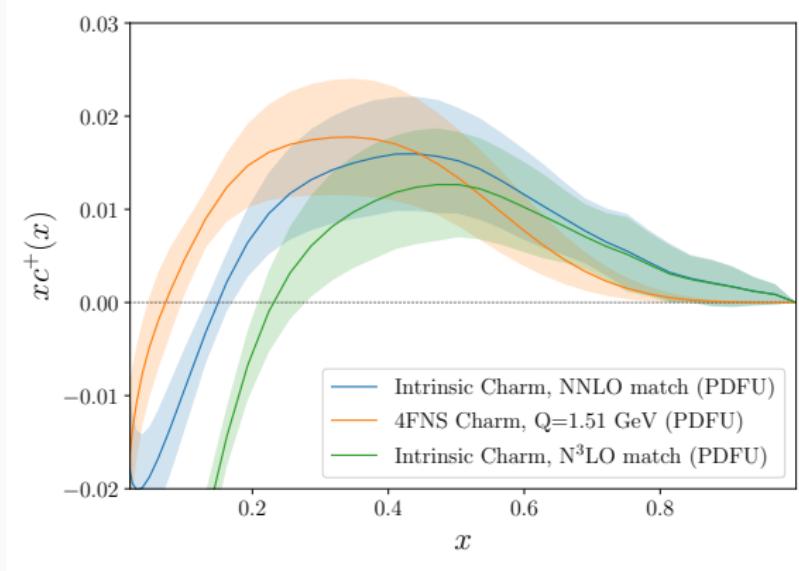
$$\tilde{\mathbf{f}}^{(n_f+1)}(\mu_{F,1}^2) = \tilde{\mathbf{E}}^{(n_f+1)}(\mu_{F,1}^2 \leftarrow \mu_h^2) \mathbf{R}^{(n_f)} \tilde{\mathbf{A}}^{(n_f)}(\mu_h^2) \tilde{\mathbf{E}}^{(n_f)}(\mu_h^2 \leftarrow \mu_{F,0}^2) \\ \times \tilde{\mathbf{f}}^{(n_f)}(\mu_{F,0}^2)$$

with $\mathbf{R}^{(n_f)}$ a flavor rotation matrix and $\tilde{\mathbf{A}}^{(n_f)}(\mu_h^2)$ the operator matrix elements (partially known up to N³LO)

for backward evolution:

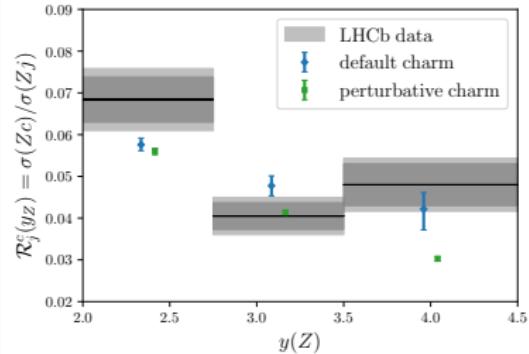
- invert $\tilde{\mathbf{E}}^{(n_f)}$: simple
- invert $\mathbf{R}^{(n_f)}$: simple
- invert $\tilde{\mathbf{A}}^{(n_f)}$: expanded or exact

Intrinsic Charm: PDF plot



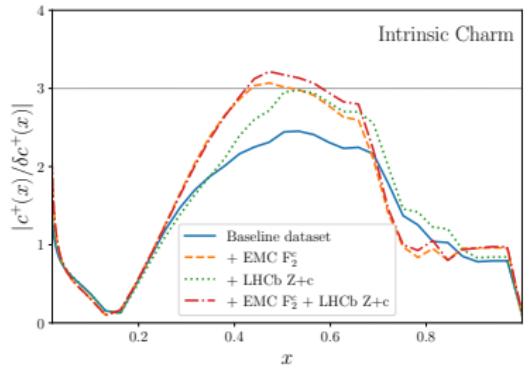
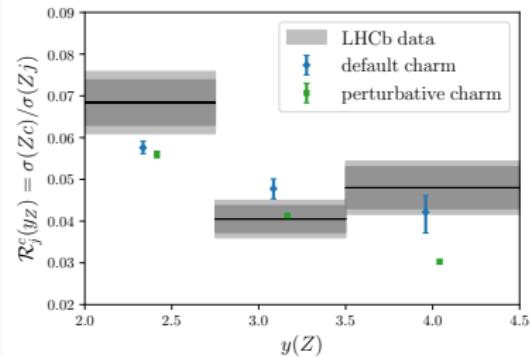
- in **3FNS** valence-like peak is still present
- for $x \leq 0.2$ the perturbative uncertainties are quite large
- the carried momentum fraction is within **1%**

Intrinsic Charm: LHCb and Significance



- match better recent **LHCb** $Z+c$ measurement [PRL128-082001]

Intrinsic Charm: LHCb and Significance



- match better recent **LHCb** Z+c measurement [PRL128-082001]
- we find a 3σ evidence of intrinsic charm
- result is **stable** with mass variation, dataset variation

3. yadism [in preparation]

yadism Physics Features



- DIS coefficient function database
- independent of boundary condition → PDF fitting
- separate features: TMC, FNS, interpolation
- constant benchmark against APFEL (eventually discovering minor bugs there)

same improvement in terms of project management as EKO!

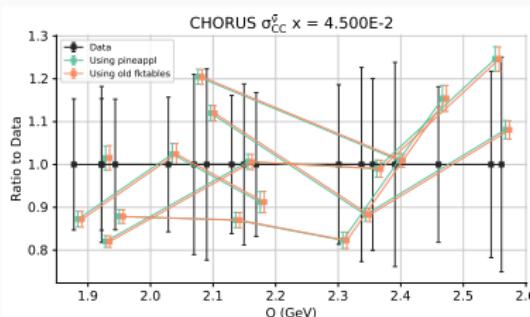
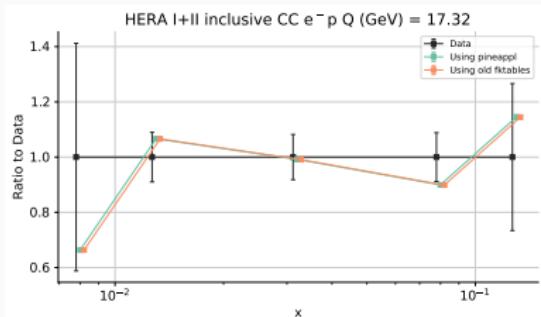
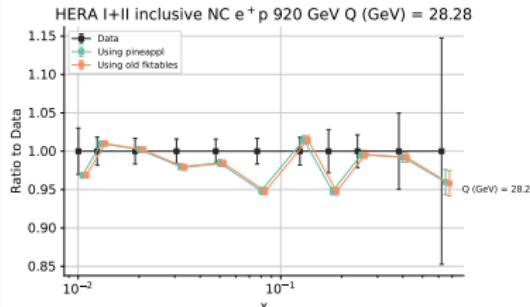
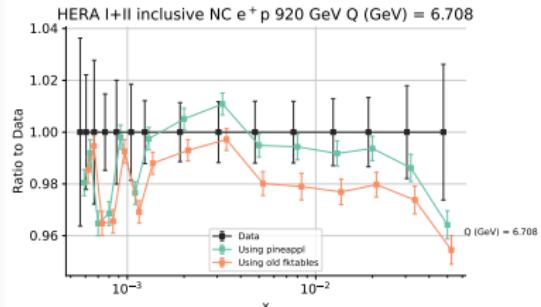
Coefficient Functions

- implemented coefficient functions:

	light	heavy	intrinsic
NC	$O(a_s^2)$ [VVM05,MVV05,MV00]	$O(a_s^2)$ [Hek19]	$O(a_s)$ [KS98]
CC	$O(a_s^2)$ [MRV08,MVV09]	$O(a_s)$ [GKR96]	$O(a_s)$ [in prep.]

- for CC intrinsic: see talk by K. Kudashkin
- implemented flavor number schemes: FFNS, ZM-VFNS, FONLL

Comparison yadism against APFEL



green, “pineappl” = yadism vs. orange, “old” = APFEL

4. Theory Prediction Pipeline

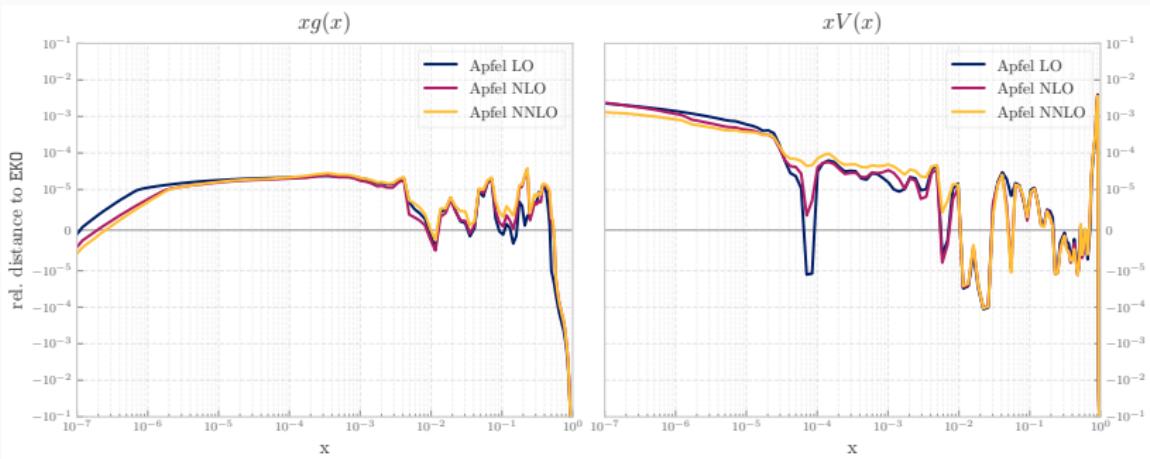
New Theory Prediction Pipeline

- We're about to develop a new pipeline for theory predictions around PineAPPL [[arXiv:2008.12789](#)]
- both, EKO and yadism, are interfaced with PineAPPL
- PineAPPL also has interfaces to mg5amc@nlo, APPLgrid, FastNLO
- aim: produce FastKernel tables used in PDF fitting

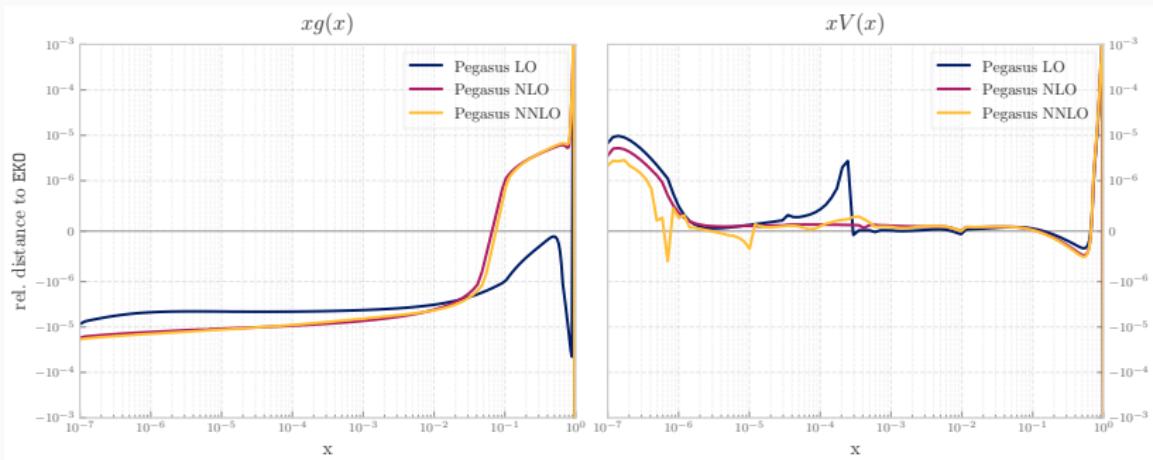
Thank you!

5. Backup slides

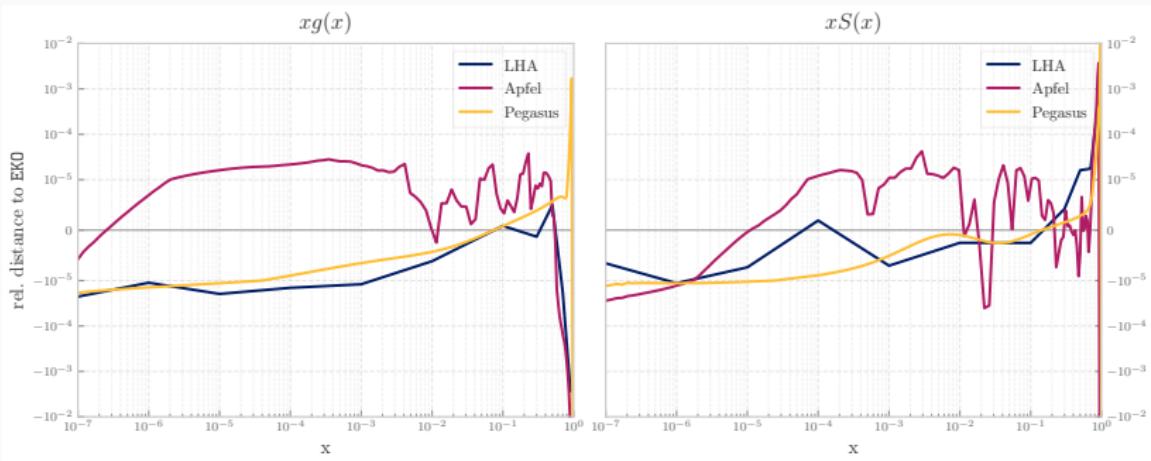
EKO APFEL benchmark



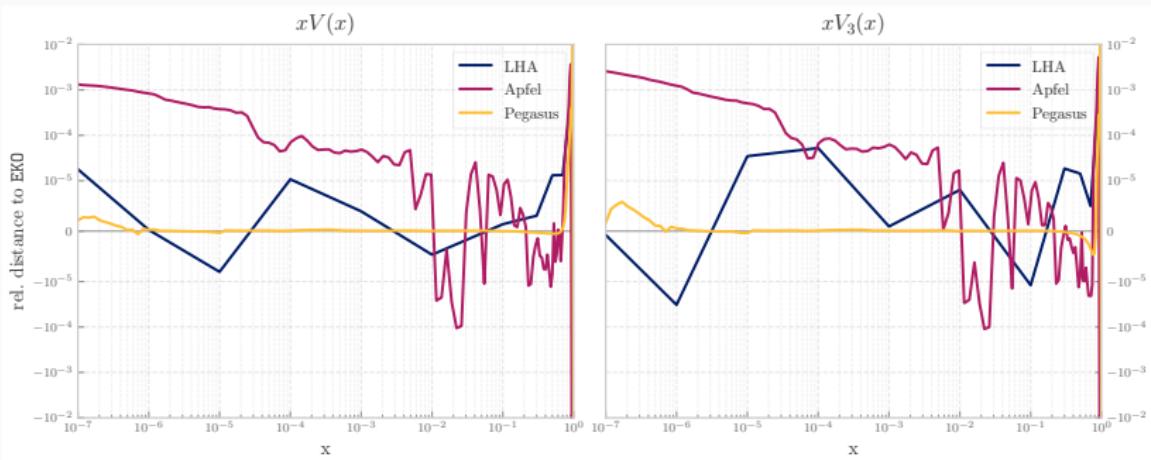
EKO PEGASUS **benchmark**



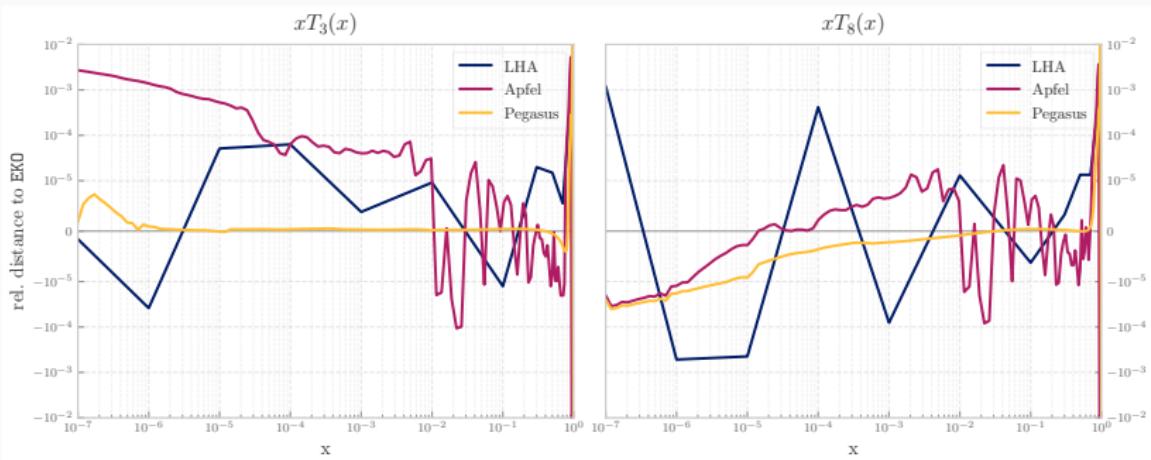
EKO LHA benchmark: g and Σ



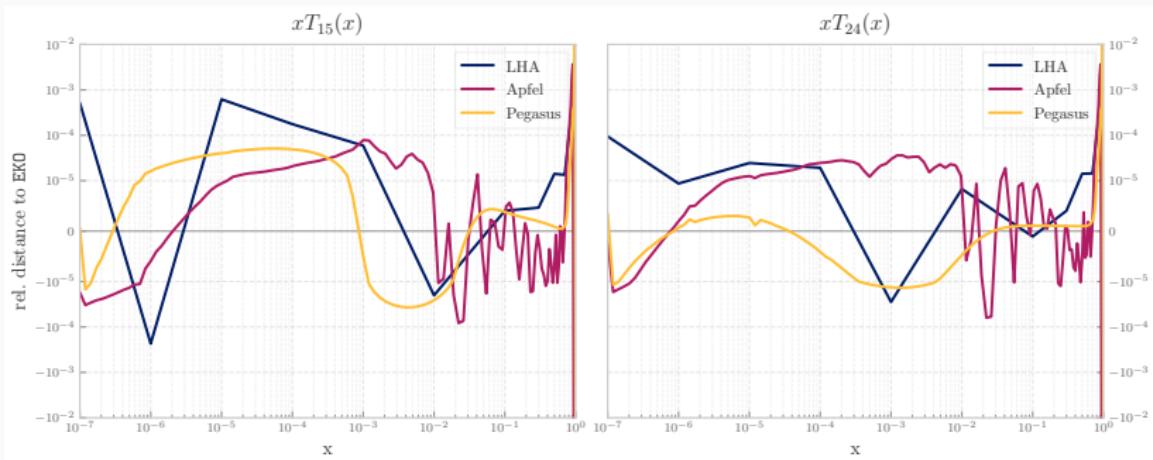
EKO LHA benchmark: V and V_3



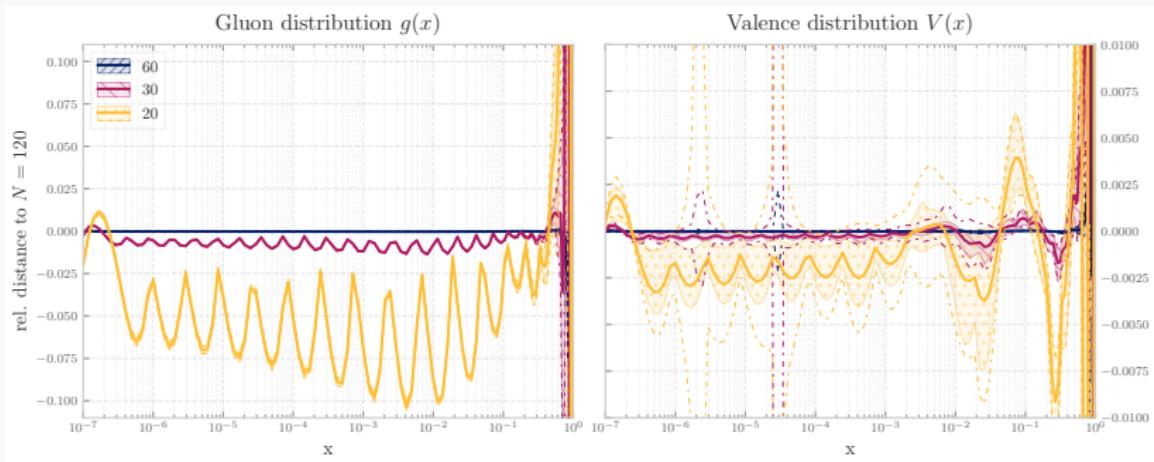
EKO LHA benchmark: T_3 and T_8



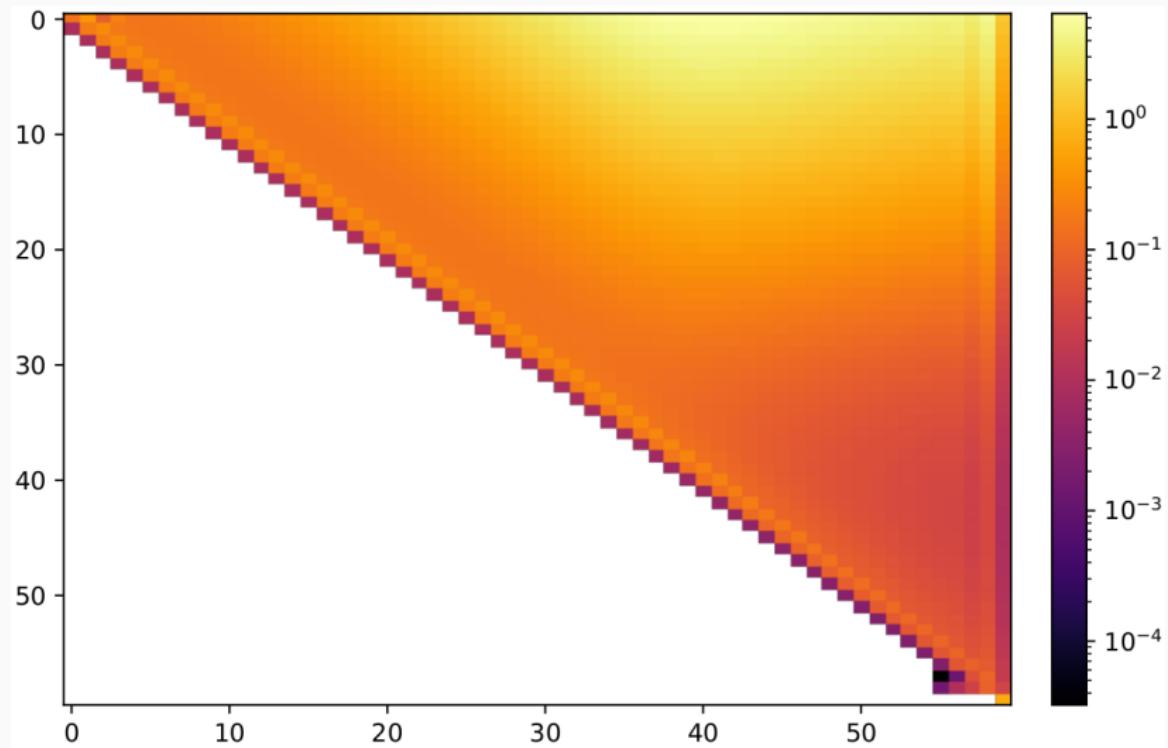
EKO LHA benchmark: T_{15} and T_{24}



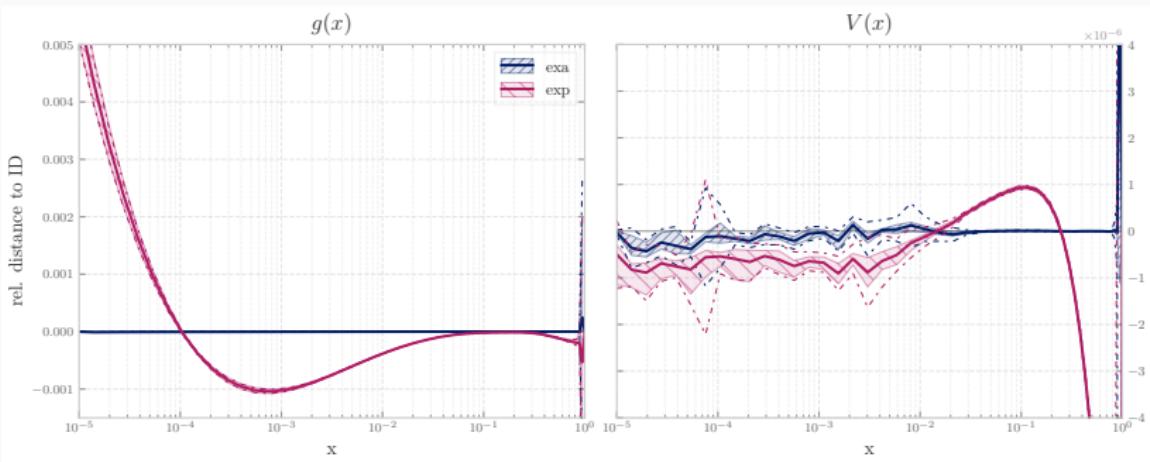
EKO Interpolation Error



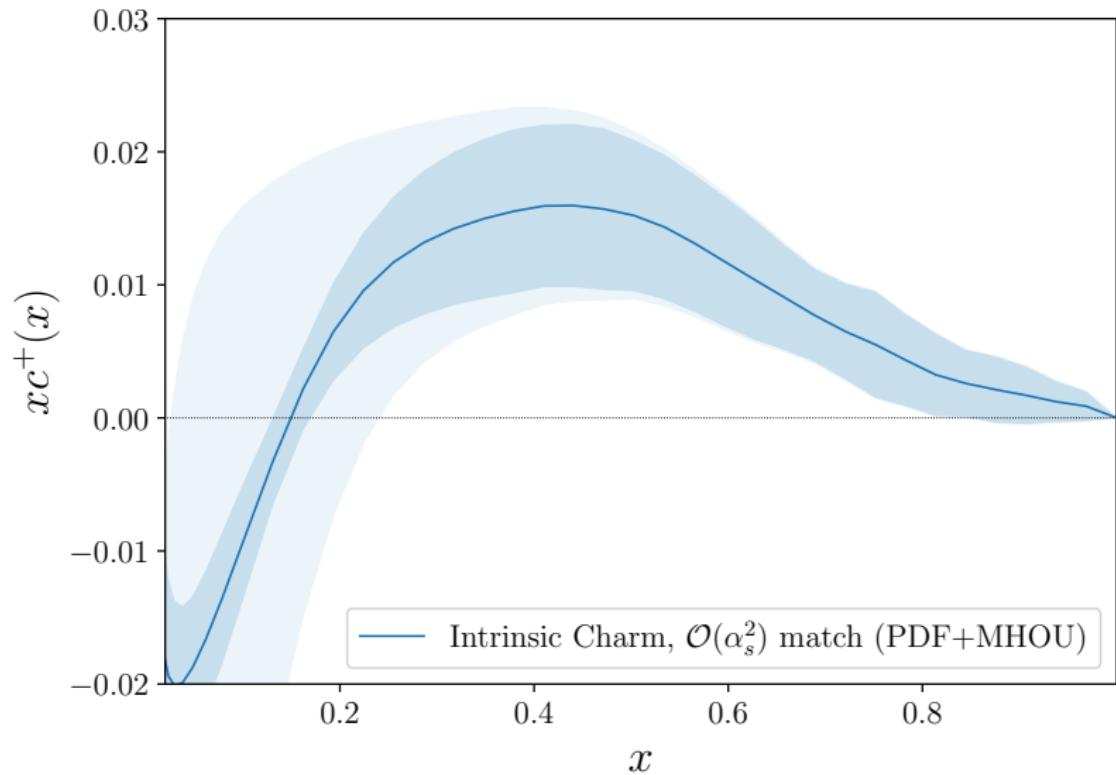
EKO Snapshot $V \leftarrow V$



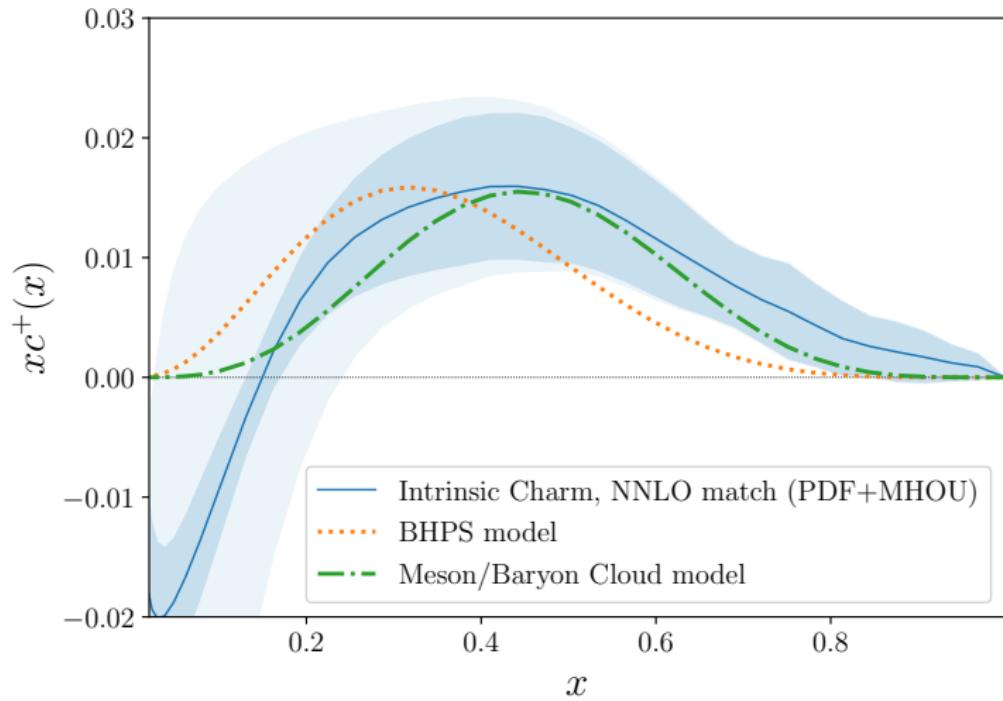
EKO Backward Evolution



IC - all uncertainties combined

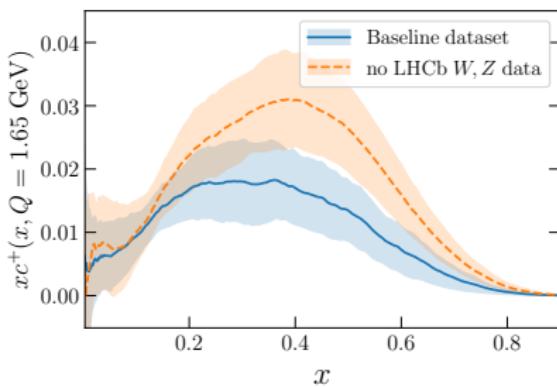
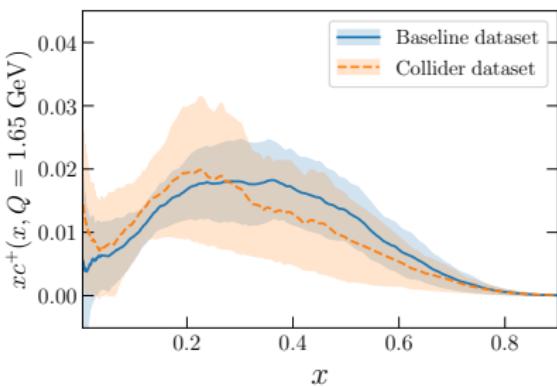
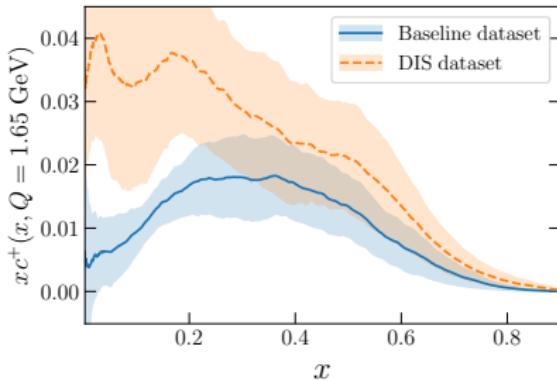
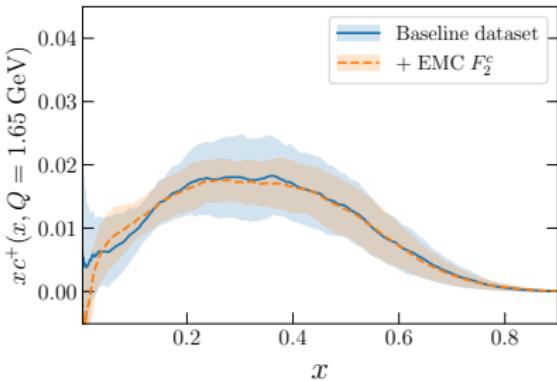


IC - model comparison

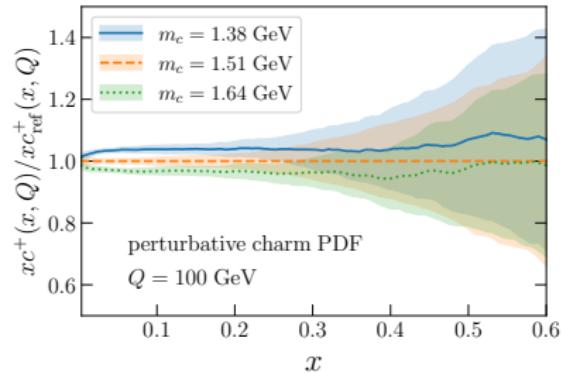
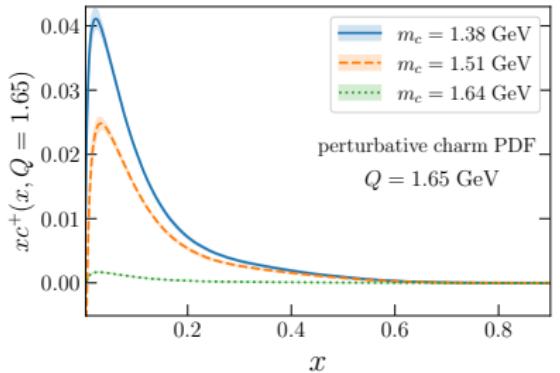
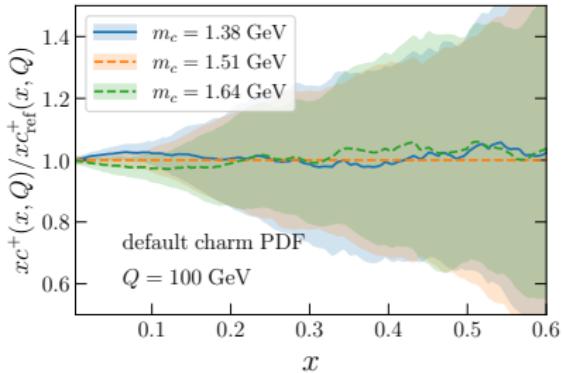
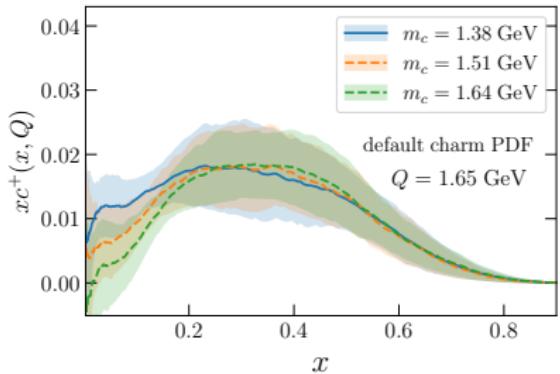


[BHPS] or [Meson/Baryon Cloud Model]

IC - dataset variation

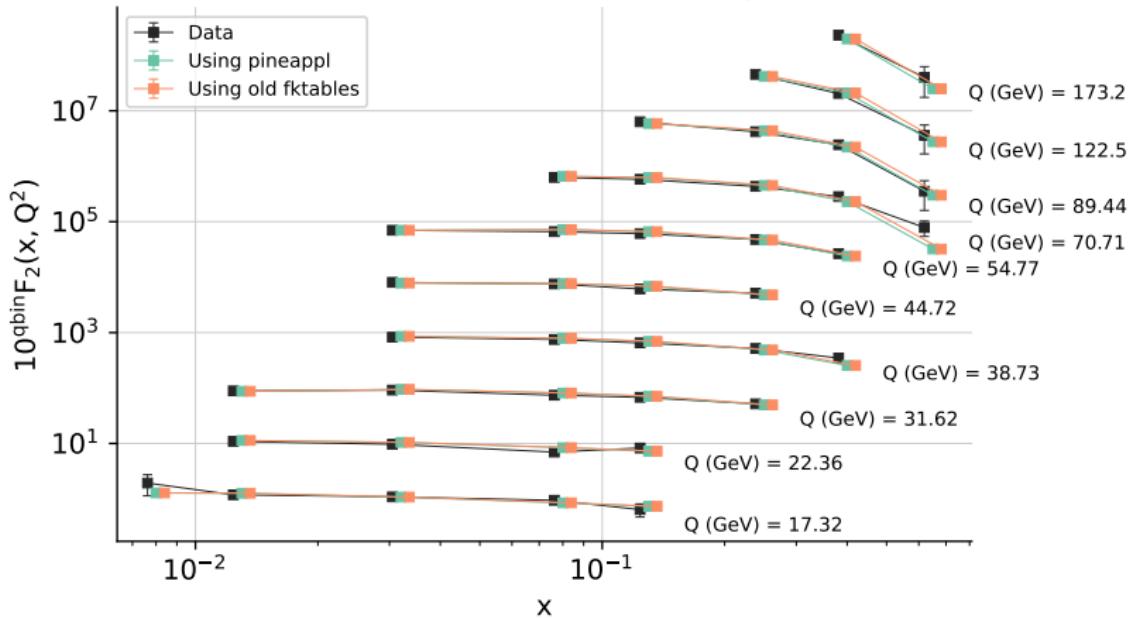


IC - mass variation



Comparison yadism against APFEL

HERA I+II inclusive CC $e^- p$



Comparison yadism against APFEL

