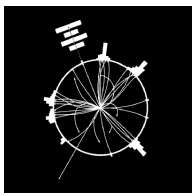


Universal Tools for High-Energy QCD Predictions

Tanjona R. Rabemananjara
rrabeman@nikhef.nl

2026-03-10

High Energy Physics Seminar
Michigan State University

Observable predictions in QCD rely heavily on the factorisation theorem. Consider the most generic process in which we have m hadrons (both **time-like** and **space-like**):

$$\mathcal{O} \sim \sum \mathcal{F}^{h_1}_{\{i_1,\ldots,i_n\}}\left(\left\{x_1^{(1)},\ldots,x_n^{(1)}\right\},\{k_T^2,\xi,\Delta\},\mu^2_{h_1},\ldots\right)\otimes\ldots\otimes\mathcal{F}^{h_m}_{\{j_1,\ldots,j_n\}}\left(\left\{x_1^{(m)},\ldots,x_n^{(m)}\right\},\{k_T^2,\xi,\Delta\},\mu^2_{h_m},\ldots\right)$$
$$\otimes\hat{O}^{(1)}_{\{i_1,\ldots,j_1\}}\left(\left\{x_1^{(1)},\ldots,x_1^{(m)}\right\},\{k_T^2,\xi,\Delta\},\{\alpha_s,\alpha_e,\ldots\}\right)\otimes\ldots$$
$$\otimes\hat{O}^{(n)}_{\{i_n,\ldots,j_n\}}\left(\left\{x_n^{(1)},\ldots,x_n^{(m)}\right\},\{k_T^2,\xi,\Delta\},\{\alpha_s,\alpha_e,\ldots\}\right)$$

Goal: Develop a **universal** framework that makes computing (upon changes of input parameters) and sharing theoretical predictions as efficient as possible $\Longleftrightarrow$ **Fast Interpolation Tables**

- **Partonic Cross-Sections:** Existing tools such as **APPLGrid** or **fastNLO** can only deal with two hadrons in the initial state
- **Non-perturbative functions:** **LHAPDF** can only deal with collinear PDFs, **TMDlib** can only deal with TMD PDFs

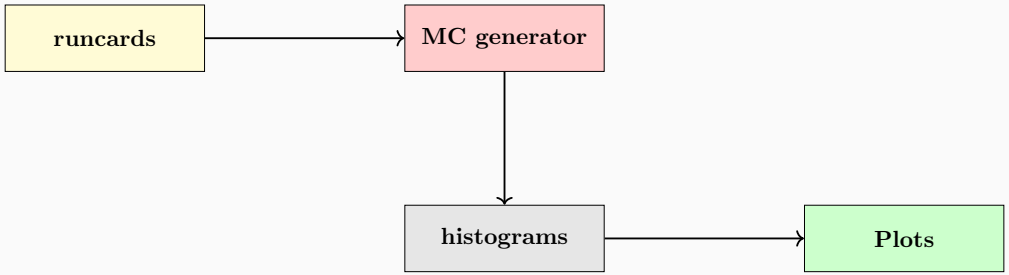
A new look at the:

1. **Short-Distance Functions**
2. **Non-Perturbative Functions**

Short-Distance Functions

Computations of theoretical predictions usually follows:

- writing runcards containing the input parameters for the process/distributions
- **Monte Carlo (MC)** generators produce histograms with PDFs and other input settings **backed-in**

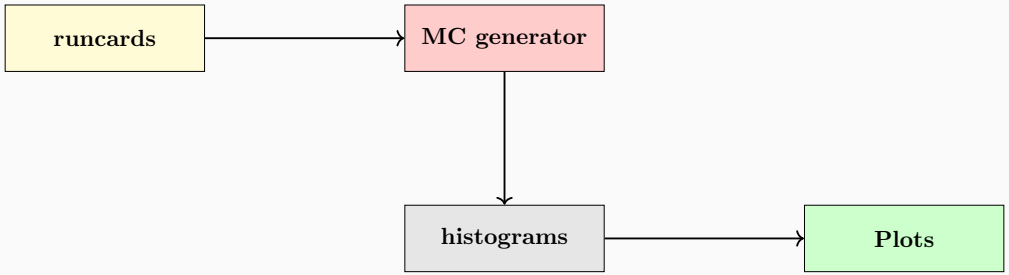


This procedure works fine!

Changing PDFs and other input settings require to rerun everything $\Longleftrightarrow$ Can become **quite expensive**

Computations of theoretical predictions usually follows:

- writing runcards containing the input parameters for the process/distributions
- **Monte Carlo (MC)** generators produce histograms with PDFs and other input settings **backed-in**



This procedure works fine!

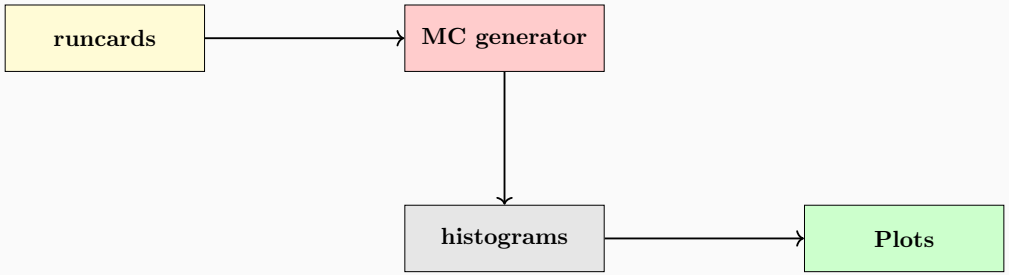
Changing PDFs and other input settings require to rerun everything $\Longleftrightarrow$ Can become **quite expensive**

Process	LO	NLO	NNLO	NNLO ($10^{-3}\delta$)
$pp \to e^+\nu_e$	0 d 1 h 56 m	0 d 3 h 43 m	114 d 6 h 18 m	24 CPU days
$pp \to e^-e^+\gamma$	0 d 17 h 55 m	1 d 19 h 48 m	1276 d 12 h 47 m	167 CPU days
$pp \to e^-\overline{\nu}_e\gamma$	0 d 22 h 18 m	3 d 16 h 59 m	1484 d 16 h 50 m	232 CPU days
$pp \to e^+\nu_e\gamma$	1 d 7 h 8 m	6 d 8 h 7 m	428 d 7 h 1 m	443 CPU days
$pp \to \mathbb{Z}$	0 d 1 h 44 m	0 d 1 h 6 m	132 d 19 h 37 m	25 CPU days
$pp \to e^-\mu^-e^+\mu^+$	0 d 5 h 43 m	0 d 4 h 32 m	219 d 16 h 33 m	45 CPU days
$pp \to e^-e^-e^+e^+$	0 d 11 h 34 m	0 d 12 h 8 m	742 d 13 h 37 m	193 CPU days
$pp \to e^-e^+\nu_\mu\overline{\nu}_\mu$	0 d 6 h 33 m	0 d 6 h 36 m	158 d 13 h 40 m	31 CPU days
$pp \to e^-\mu^+\nu_\mu\overline{\nu}_e$	0 d 13 h 33 m	1 d 22 h 9 m	521 d 2 h 20 m	119 CPU days
$pp \to e^-e^+\nu_e\overline{\nu}_e$	0 d 23 h 36 m	0 d 17 h 46 m	270 d 6 h 59 m	52 CPU days

MATRIX runtime [Grazzini et al., 2018]

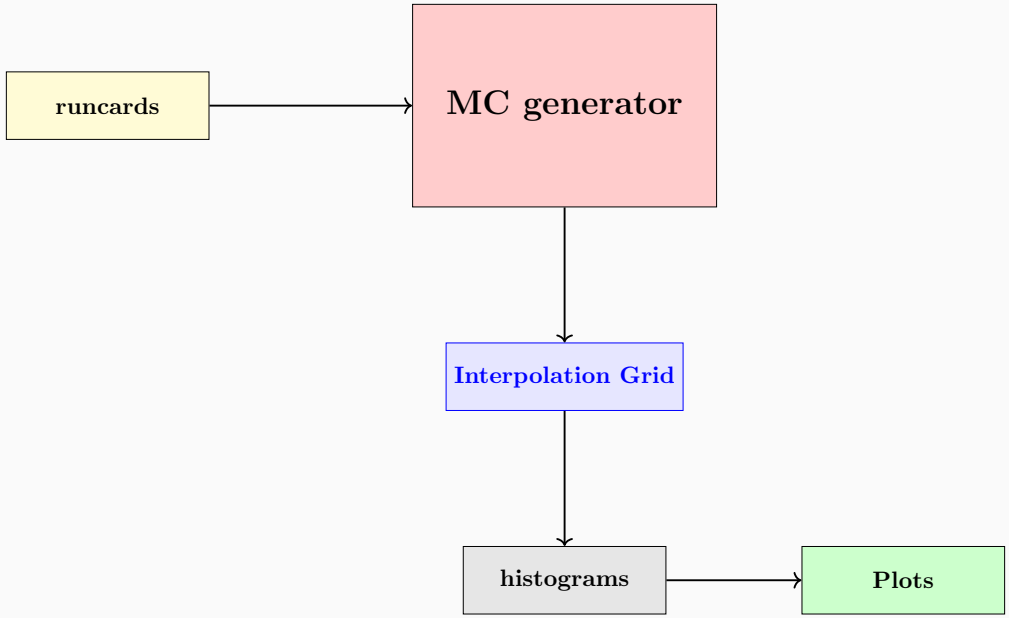
Computations of theoretical predictions usually follows:

- writing runcards containing the input parameters for the process/distributions
- **Monte Carlo (MC)** generators produce histograms with PDFs and other input settings **backed-in**

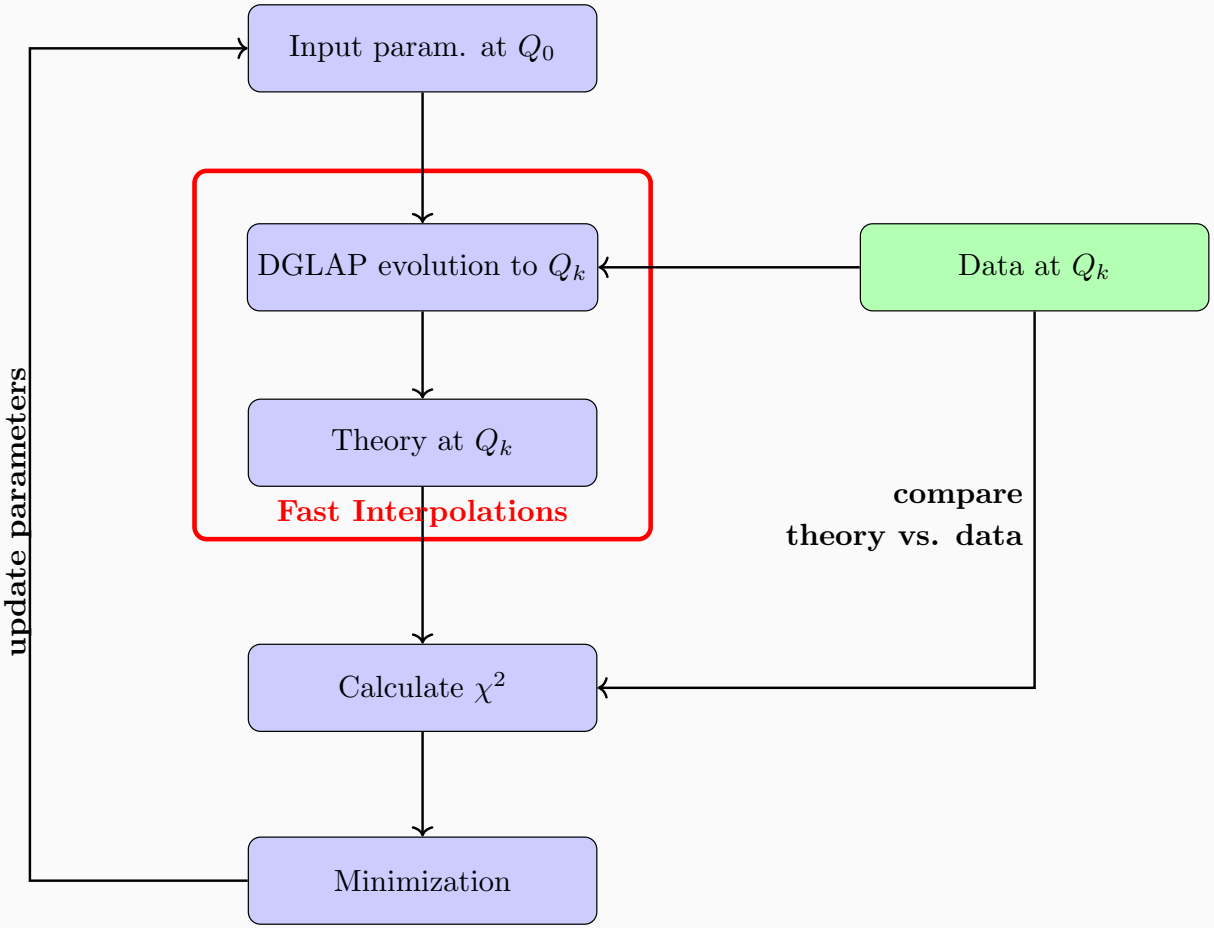


This procedure works fine!

Changing PDFs and other input settings require to rerun everything $\Longleftrightarrow$ Can become **quite expensive**



Grids can be convolved with arbitraty non-perturbative inputs and different settings in a **matter of seconds**



DGLAP Evolution Equation:

$$f_i(x_\alpha, Q_k) = E_{ij,\alpha,\beta}(Q_k \leftarrow Q_0) f_j(x_\beta, Q_0)$$

Both the **Evolution** and the **Theoretical Predictions** at Q_k can be **pre-computed** using **Fast Interpolation Grids (FK Tables)**.

$$\sigma = \sum_i \sum_a f_a(x_i; Q_0^2) \mathrm{FK}_a(x_i; Q_0^2)$$

$$\mathrm{FK}_a(x_i; Q_0^2) = \sum_{b,j,k,l} \alpha_s^{n+k}(Q_j^2) \mathrm{EKO}_{a,i}^{b,l,j} \sigma_b^k(x_l, Q_j^2)$$

- **Compute once and re-use many times**: avoids unnecessary repeated computations when exploring different inputs
- **Enables (global) extraction of non-perturbative functions and uncertainty propagation feasible**
- **Allows cheap and systematic exploration of scale/theory variations**
- **Provides numerical stability and reproducibility**: removes stochastic noises in MC, produce deterministic predictions
- **Makes theoretical predictions accessible and usable across collaborations, theorists, and experimentalists**

the APPLgrid project

- **Limited support for EWK corrections and/or Mixed QCD-EWK orders**: was designed primarily for (NLO) QCD calculations
- **Less flexible treatment of perturbative orders and couplings**: less flexible recombination of contributions for convolutions
- **Memory and performance limitations**: attempts to extend APPLgrid to include additional physics contributions ran into memory and performance issues

fastNLO

- **No support for EWK corrections**: lacks native support for EWK corrections and/or mixed QCD-EWK perturbative orders
- **Complicated interface and reduced flexibility**: for e.g. the user is limited to a number of combined (μ_R, μ_F) choices
- **More restricted physics coverage**: traditionally tailorer mainly to jet observables and certain QCD processes, with **specific generator interfaces**

LO:	$\mathcal{O}(\alpha_s^2\alpha^0)$ $\boxed{C_{000}}$	$\mathcal{O}(\alpha_s^1\alpha^1)$ $\boxed{C_{000}}$	$\mathcal{O}(\alpha_s^0\alpha^2)$ $\boxed{C_{000}}$		
NLO:	$\mathcal{O}(\alpha_s^3\alpha^0)$ $\boxed{C_{000} + C_{100}\,\ell_R + C_{010}\,\ell_F + C_{001}\,\ell_D}$	$\mathcal{O}(\alpha_s^2\alpha^1)$ $\boxed{\cdots}$	$\mathcal{O}(\alpha_s^1\alpha^2)$ $\boxed{\cdots}$	$\mathcal{O}(\alpha_s^0\alpha^3)$ $\boxed{\cdots}$	
NNLO:	$\mathcal{O}(\alpha_s^4\alpha^0)$ $\boxed{\begin{aligned} &C_{000} + C_{100}\ell_R + C_{010}\ell_F + C_{001}\ell_D \\ &\quad + C_{200}{\ell_R}^2 + C_{020}{\ell_F}^2 + C_{002}{\ell_D}^2 \\ &\quad + C_{110}\ell_R\ell_F + C_{101}\ell_R\ell_D + C_{011}\ell_F\ell_D \end{aligned}}$	$\mathcal{O}(\alpha_s^3\alpha^1)$ $\boxed{\cdots}$	$\mathcal{O}(\alpha_s^2\alpha^2)$ $\boxed{\cdots}$	$\mathcal{O}(\alpha_s^1\alpha^3)$ $\boxed{\cdots}$	$\mathcal{O}(\alpha_s^0\alpha^4)$ $\boxed{\cdots}$

- Initially developed to support DIS and DY-like processes with support for **Electroweak Corrections** and explicit dependence on the **orders**, **channels**, and **scales**
- Extended (in **v1**) to support generic processes that involve any arbitrary number of **convolutions** & **kinematics**: **Single Parton Scattering** or **SPS** (PDFs, TMDs, GTMDs, GPDs, ...), **DPS**, **TPS**, ...

- PineAPPL is already interfaced to a **wide range** of MC generators and used in various HEP studies:
  - **Yadism**: inclusive Deep Inelastic Scattering (DIS) processes [Candido et al., **2024**]
  - **APFEL++**: inclusive DIS, Single Inclusive Annihilation (SIA), Semi-Inclusive DIS (SIDIS) [Bertone, **2018**; Bertone et al., **2014**]
  - MadGraph5_aMC@NLO [Carrazza et al., **2020**]
  - **NNLOJet** (**PineJet**): LHC W/Z, inclusive jets, di-jets, … [Cruz-Martinez et al., **2025**]
  - **MATRIX**: top quark (pair) productions, … [Devoto et al., **2025**]
  - **MCFM**: polarized W productions (for polarised DY) [Boughezal et al., **2021**]
  - SHERPA + MCgrid / Rivet
  - **xFitter**
  - **NNPDF**
  - **nCTEQ** (Open Heavy Flavour in GM-VFNS)
  - **ABMP-MATRIX**
  - …

To generate **PDF-independent** predictions, we use the **Lagrange Interpolation** to represent the PDFs:

$$f_a(x_1, Q^2) f_b(x_2, Q^2) \approx \sum_{i,j,k} f_a(x_i, Q_k^2) f_b(x_j, Q_k^2) L_i(x_1) L_j(x_2) L_k(Q^2)$$

with Lagrange Polynomials L_i over the 3D Grid $\left\{\left(x_i, x_j, Q_k^2\right)\right\}_{\{i,j,k\}}$. Using the master factorisation formula:

$$\begin{aligned}\frac{d\sigma}{d\mathcal{O}} &= \sum_{a,b} \int_0^1 dx_1 \int_0^1 dx_2 \int_{Q^2_{\min}}^{Q^2_{\max}} dQ^2 f_a(x_1, Q^2) f_b(x_2, Q^2) \frac{d\hat{\sigma}_{ab}}{d\mathcal{O}}(x_1, x_2, Q^2, \mathcal{O}) \\ &= \sum_{a,b} \sum_{i,j,k} \sum_{m,n} f_a(x_i, Q_k^2) f_b(x_j, Q_k^2) \alpha_s^{m(Q^2)} \alpha^n \frac{d\Sigma_{abijkmn}}{d\mathcal{O}}\end{aligned}$$

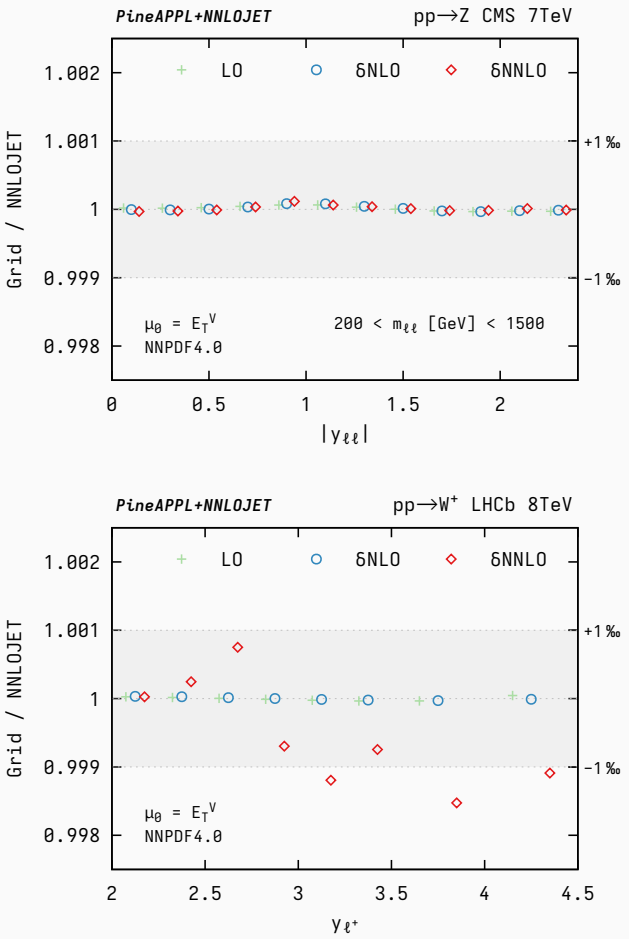
where the partonic part can now be writen as:

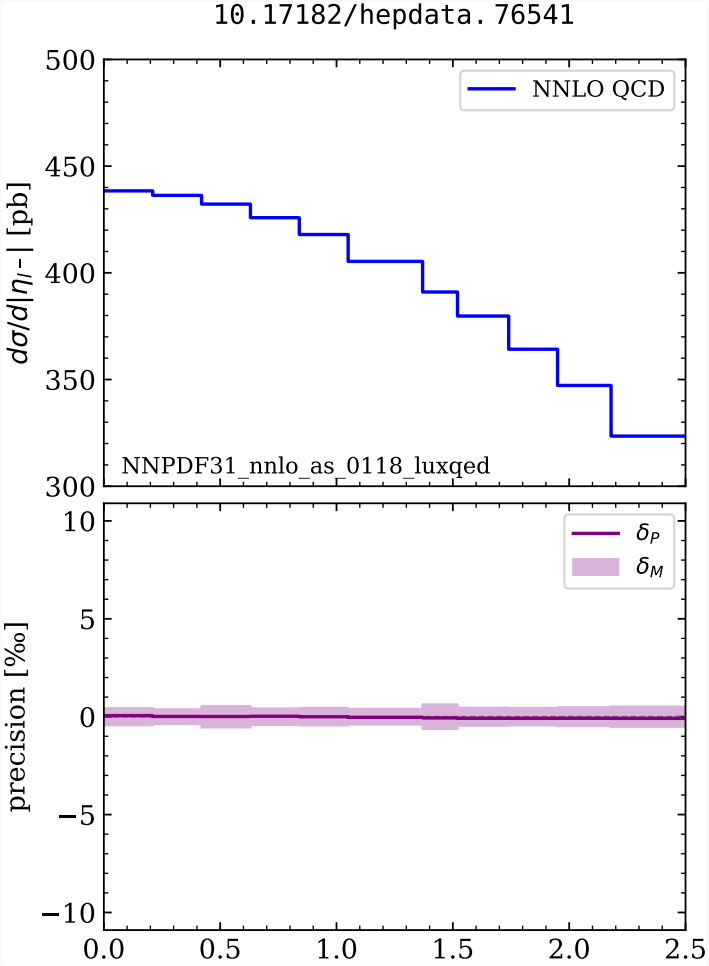
$$\frac{d\Sigma_{abijkmn}}{d\mathcal{O}} = \int_0^1 dx_1 \int_0^1 dx_2 \int_{Q^2_{\min}}^{Q^2_{\max}} dQ^2 L_i(x_1) L_j(x_2) L_k(Q^2) \frac{d\sigma_{ab}^{i,k}}{d\mathcal{O}}(x_1, x_2, Q^2, \mathcal{O})$$

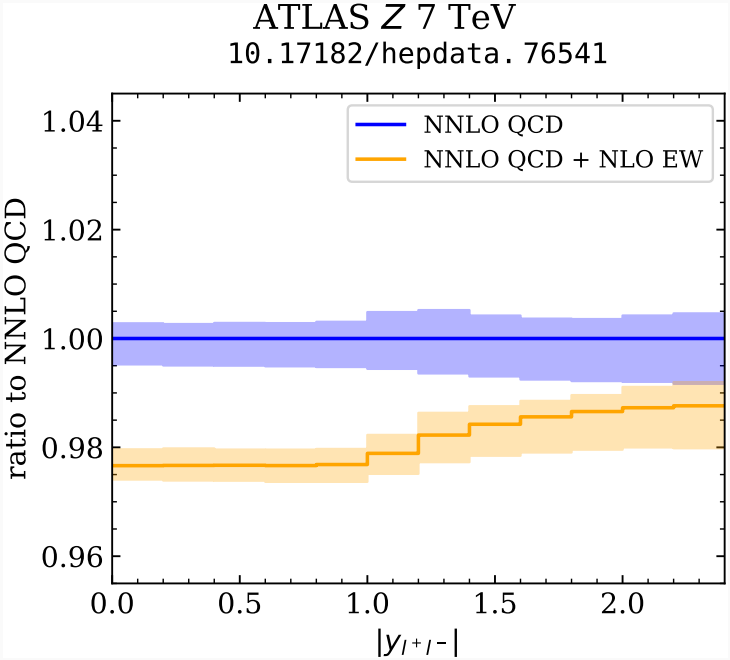
This is the object that we represent in the Grid and allows performing **extremely fast convolutions** off-line

Process class	Processes
$e + e^-$ scattering Jet production	$e + e - \to 2$ jets
	$e + e - \to 3$ jets
ep scattering Jet production	$ep \to \ell + 1$ jet
	$ep \to \ell + 2$ jets
pp scattering Jet production	p p $\longrightarrow$ 1 jet + X
	p p $\longrightarrow$ 2 jets
Vector boson (+ jet) production	$pp \to (\gamma^*, Z) + 0$ jet
	$pp \to (\gamma^*, Z) + 1$ jet
	$pp \to W^\pm + 0$ jet
	$pp \to W^\pm + 1$ jet
Photon (+ jet) production	$pp \to \gamma + X$
	$pp \to \gamma + 1$ jet
	$pp \to \gamma\gamma$
Higgs (+ jet) production	$pp \to H + X$
	$pp \to H + 1$ jet

[Cruz-Martinez et al., 2025; Huss and others, 2025]







The effects of **Electroweak correctins can be sizeable** and can exceed the NNLO QCD uncertainties (7 point variations)

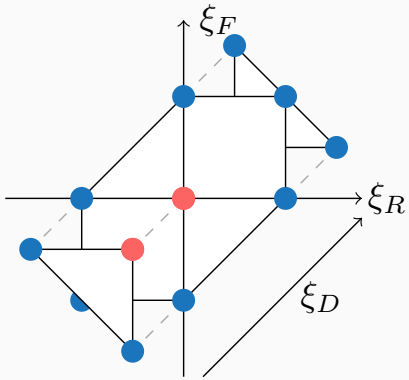
[Devoto et al., **2025**]

PineAPPL retains the explicit dependence on the unphysical scales:

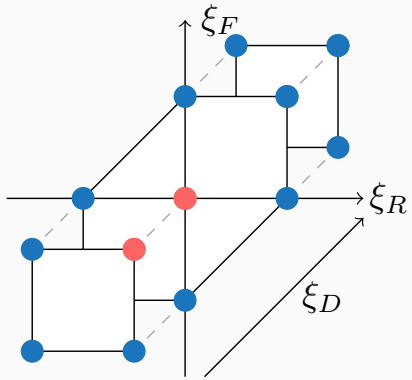
$$\Sigma_{\mathcal{O}}^{n,m} = \sum_{i,j,k} C_{ijk}^{n,m} \ell_R^i \ell_F^j \ell_D^k \quad \text{where} \quad \ell_X = \ln\left(\frac{\mu_X^2}{Q^2}\right)$$

which makes it suitable for investigating various scale-variation prescriptions.

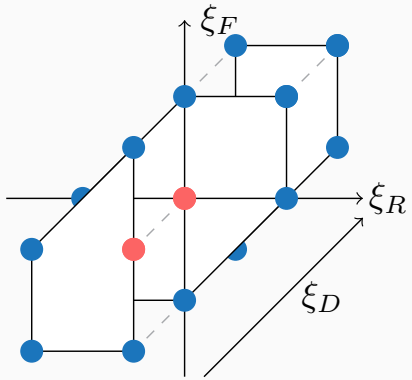
In addition to the standard 5-, 7-, and 9-point variations, PineAPPL implements additional scale variation prescriptions in the case Fragmentation scale is present.



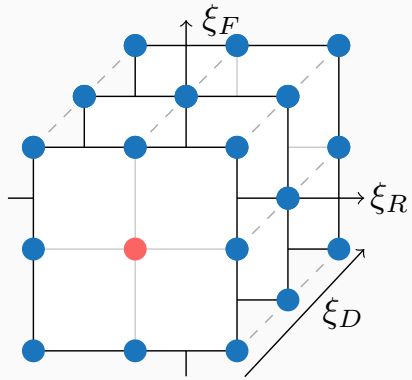
13-point



15-point



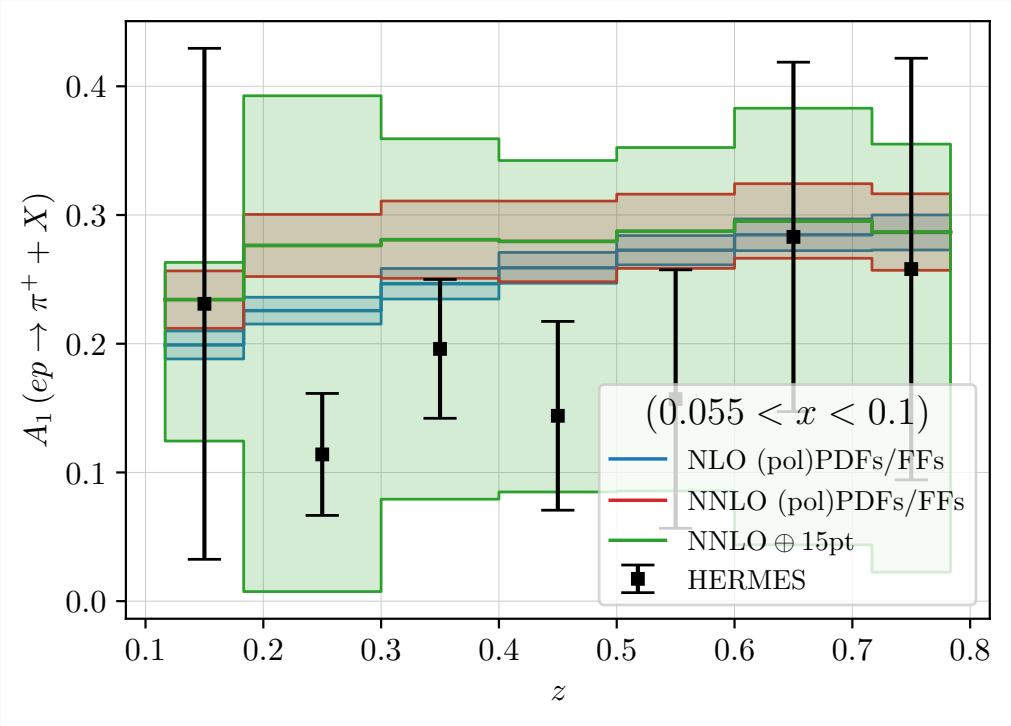
17-point



27-point

Example of single-inclusive hadron production in **Semi-Inclusive Deep Inelastic Scattering (SIDIS)**:

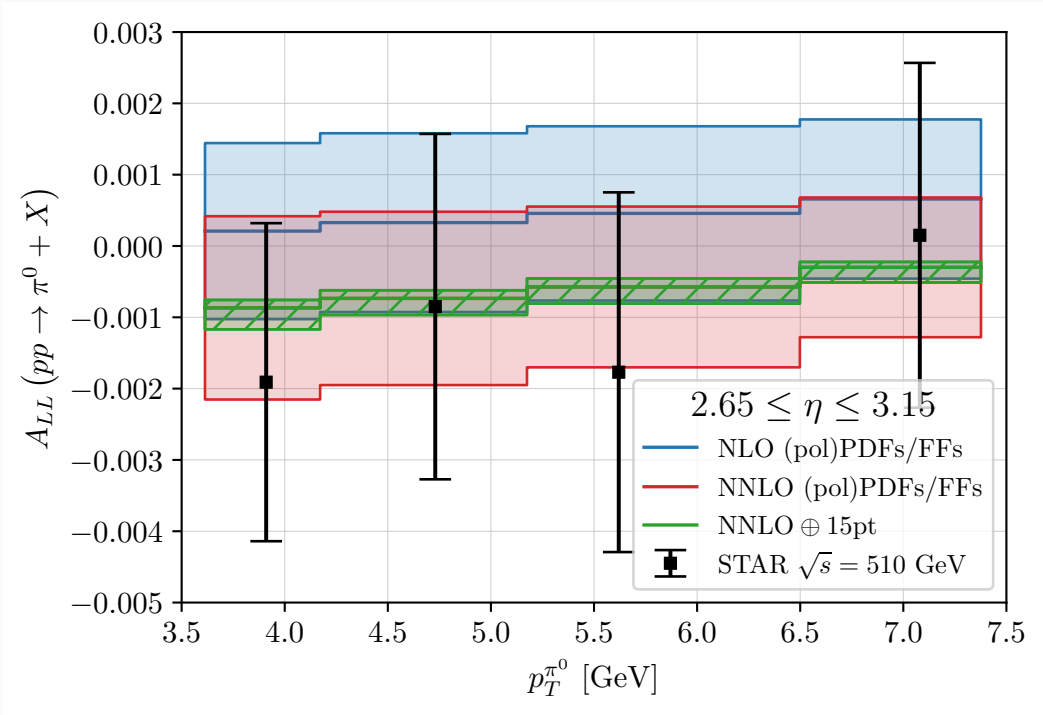
$$A_1(ep \to \pi^+ + X) \propto \frac{\text{Polarized PDF} \otimes \text{Unpolarized FF}}{\text{Unpolarized PDF} \otimes \text{Unpolarized FF}}$$

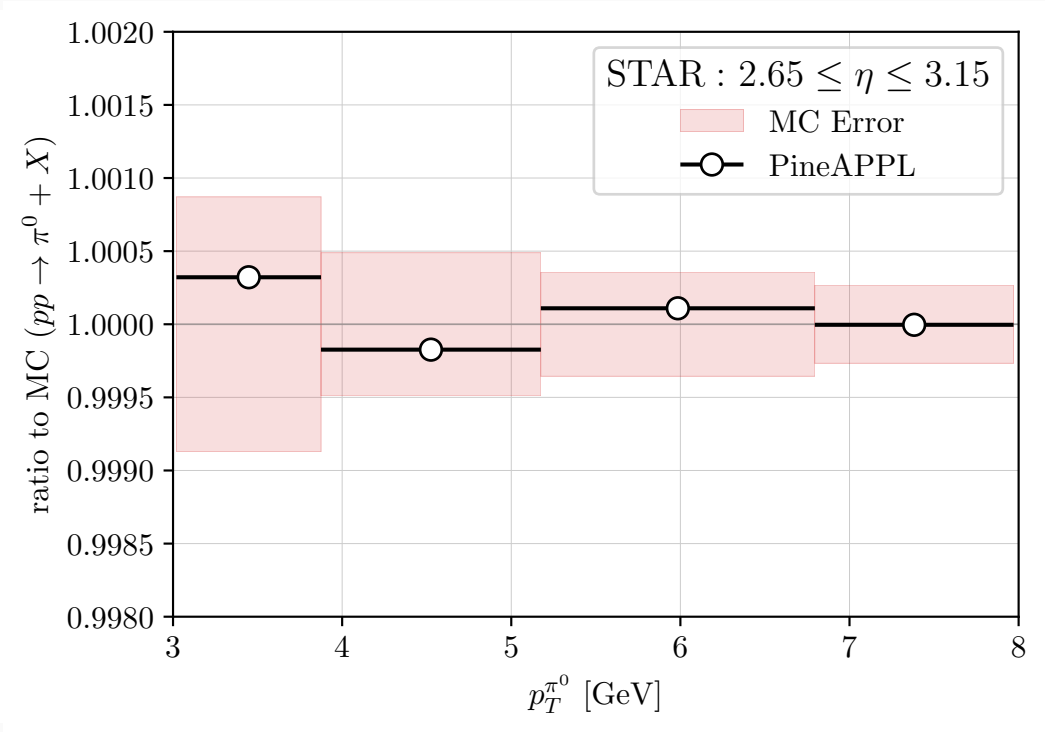


- **Photo-absorption asymmetry** as measured by HERMES [Airapetian and others, **2019**] during the 1996-2000 data-taking
- MEs computed consistently at NLO and NNLO; non-perturbative uncertainties come from the Polarized PDFs
- Missing Higher Order Uncertainties (**MHOUs**) are large $\iff$ Need to be **systematically** included

Example of single-inclusive hadron-fragmentation in proton-proton collisions [Adam and others, 2018; Jager et al., 2003]:

$$A_{LL}(pp \to \pi^0 + X) \propto \frac{\text{Polarized PDF} \otimes \text{Polarized PDF} \otimes \text{Unpolarized FF}}{\text{Unpolarized PDF} \otimes \text{Unpolarized PDF} \otimes \text{Unpolarized FF}}$$





PineAPPL provides CLI interfaces to convert APPLgrid and fastNLO grids into the PineAPPL format.

```
pineappl import applfast-atlas-dijets-v2-fc-fnlo-arxiv-1312.3524-xsec005.tab.gz grid.pineappl.lz4 NNPDF40_nnlo_as_01180

######################################################################################
#
# fastNLO_toolkit
# Version 2.5.0_2826
#
# C++ program and toolkit to read and create fastNLO v2 tables and
# derive QCD cross sections using PDFs, e.g. from LHAPDF

b   PineAPPL     fastNLO      rel. diff     svmaxreldiff
-+------------+------------+-------------+-------------
0  7.1408257e1  7.1408257e1  2.8865799e-15  9.9920072e-15
1  1.3299309e1  1.3299309e1 -2.4424907e-15  9.9920072e-15
2  1.2727839e0  1.2727839e0 -8.8817842e-15 -1.9206858e-14
3 2.3961127e-2 2.3961127e-2 -2.8865799e-15 -1.6653345e-14
```

- **Repository**: https://github.com/NNPDF/pineappl
- **CLI**: https://github.com/NNPDF/pineappl/blob/master/docs/cli-tutorial.md
- **C/C++ Examples**: https://github.com/NNPDF/pineappl/tree/master/examples/cpp
- **Python Examples**: https://pineappl.readthedocs.io/en/latest/tutorials.html

What is PineAPPL?

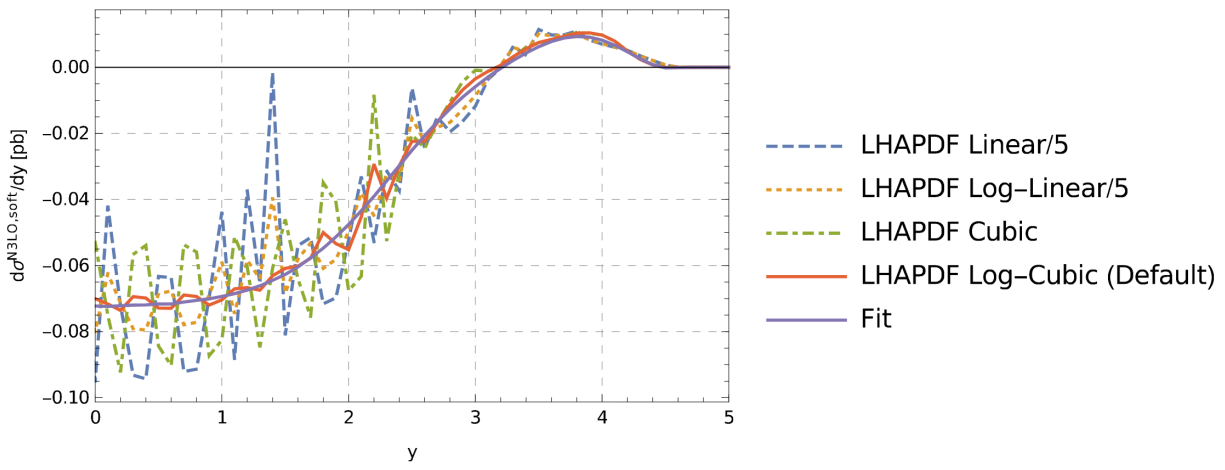
This repository contains programs, libraries and interfaces to read and write `PineAPPL` interpolation grids, which store theoretical predictions for high-energy collisions independently from their PDFs and the strong coupling.

PineAPPL grids are generated by Monte Carlo generators, and the grids in turn can be convolved with PDFs to produce tables and plots, such as the following one:

Non-Perturbative Functions

- **LHAPDF is difficult to integrate within Projects**

- **The accuracy of LHAPDF interpolation is insufficient for high-precision calculations**: artificial oscillations in differential N3LO Higgs [Dulat et al., **2018**], instabilities when computing the log-derivative of the PDFs [Ball et al., **2016**]



- **Loading and Interpolation Evaluation time could become a bottleneck**: an improvement of a factor of 2 is already a significant improvement for Monte Carlo and Parton Shower codes
- **Does not support interpolations across additional parameters**: such as the atomic mass number A (for nuclear PDF fits & studies) and strong coupling $\alpha_s(M_Z)$
- **Can only support collinear PDFs**: cannot be used to study TMDs, GPDs, GTMDs, ...

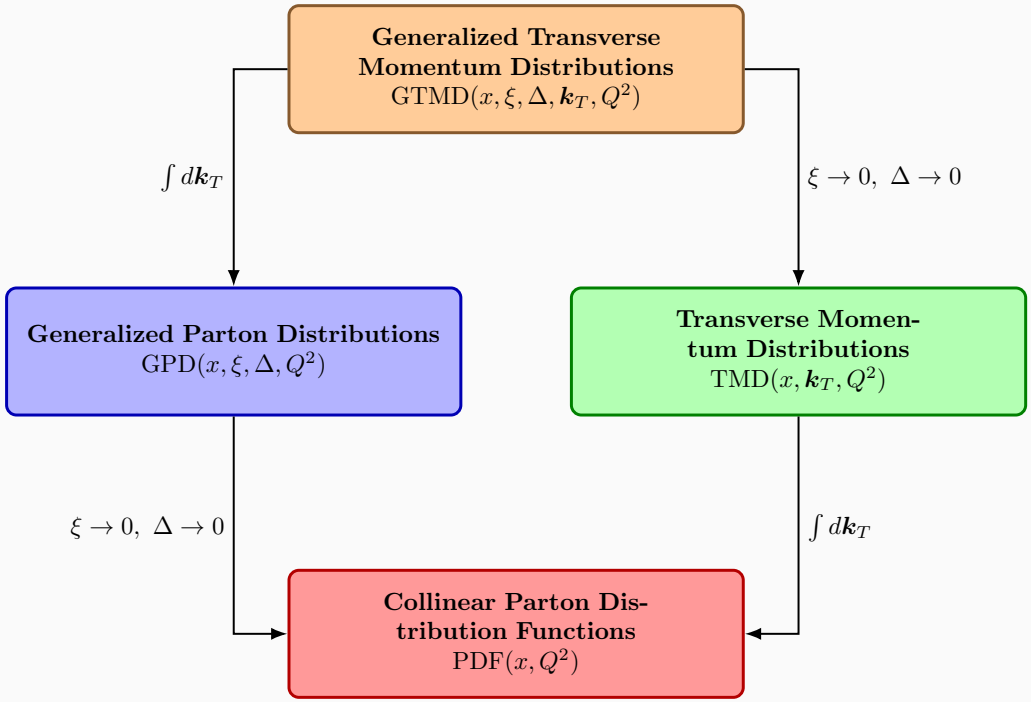
NeoPDF aims to provide a unified and coherent framework for analysing generic non-perturbative distribution functions.

Generally, the object that NeoPDF describes is of the form:

$$f_i(x, k_T^2, \xi, \Delta, \mu^2; \alpha_s(M_Z), A)$$

NeoPDF's core features:

- A modern subgrid-based design to ensure flexibility, efficiency, and scalability
- Faster data loading and interpolation evaluation time
- More accurate and precise interpolation based on **Chebyshev Polynomials** [Diehl et al., **2022**]
- A dedicated file format, specifically engineered to overcome the rigidity and inefficiencies of legacy libraries
- A Command Line Interface (CLI) tool to perform various operations/interpolations on PDF sets



A **Subgrid** object is generally represented as a rank-8 tensor (inc. flavour dependence):

$$\mathfrak{S}_{i,j,k,\ell,m,n,o} = \mathfrak{S}\left(A_i, \alpha_{s,j}, k^2_{T,k}, \xi_{\{\ell\}}, \Delta_m, x_n, Q^2_o\right)$$

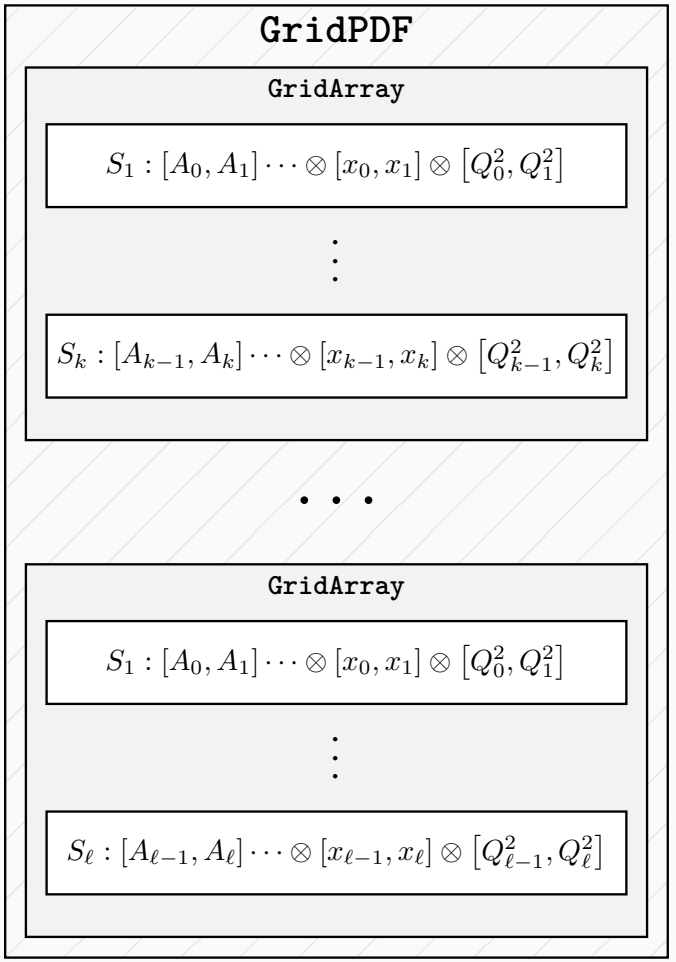
which, depending on the distributions, reduces to lower-dimensional arrays. For e.g., for the standard case of proton PDF, it reduces to 2D array $\mathfrak{S}_{m,n}$.

In the simplest case of 1D array, a Grid member is a conjunction of k subgrids:

$$[z_0, z_1, ..., z_k]_{n_1,n_2,...,n_k}$$

In the general case, a subgrid S_j is a **hyperrectangle** given by the **Cartesian** product of the intervals:

$$S_j = [A_{j-1}, A_j]_{n_j} \otimes ... \otimes [x_{j-1}, x_j]_{n_j} \otimes [Q^2_{j-1}, Q^2_j]_{n_j}$$



NeoPDF combines **binary serialization** and **LZ4 compression**, prioritising compactness, flexibility, scalability, and efficient access patterns over **human readable format** $\Longleftrightarrow$ **Significantly reduces storage requirements**

The NeoPDF data compression format is formally represented as a mapping:

$$\mathcal{F} = \left\{ \mathcal{H}, \mathcal{O}, \bigcup_{m=0}^{N_m} \mathcal{G}_m \right\},$$

where $\mathcal{H}$ is the metadata header, $\mathcal{O}$ the offset table, $\mathcal{G}_m$ the compressed data block for member m, and the offset table $\mathcal{O}$ is a **bijection**:

$$\mathcal{O} : m \longmapsto \text{addr}(\mathcal{G}_m\ )$$

that maps a member m to the address of the corresponding grid.

PDF Set	Nb. Members	LHAPDF/ TMDlib	NeoPDF
PDF4LHC21	40	31 MB	16 MB
NNPDF4.0	100	158 MB	85 MB
NNPDF4.0	1000	1.55 GB	830 MB
nNNPDF3.0$\oplus$	19$\times$200	3.20 GB	1.43 GB
MAP22 FF	250	2.50 GB	930 MB

Table 1: File size comparisons

The **Chebyshev polynomials** of the **first kind** is expressed as:

$$T_k(\cos\theta) = \cos(k\theta)$$

For a sufficiently smooth function $f(t)$, it can be expanded in the Chebyshev series as:

$$f(t) = \lim_{N\to\infty} f_N(t), \quad f_N(t) = \sum_{k=0}^{N} a_k T_k(t)$$

The coeffs a_k are numerically difficult to evaluate $\Longrightarrow$ construct the Chebyshev interpolants using the **Barycentric Formula**:

$$p_N(u) = \sum_{j=0}^{N} f(u_j)\ell_j(u)$$

where the basis functions ℓ_j given on a set of distinct Chebyshev nodes is:

$$\ell_j(u) = \frac{\lambda_j}{u-u_j}\left(\sum_{i=0}^{N}\frac{\lambda_i}{u-u_i}\right)^{-1}, (\lambda_i)^{-1} = \prod_{j\neq i}(u_j-u_i).$$

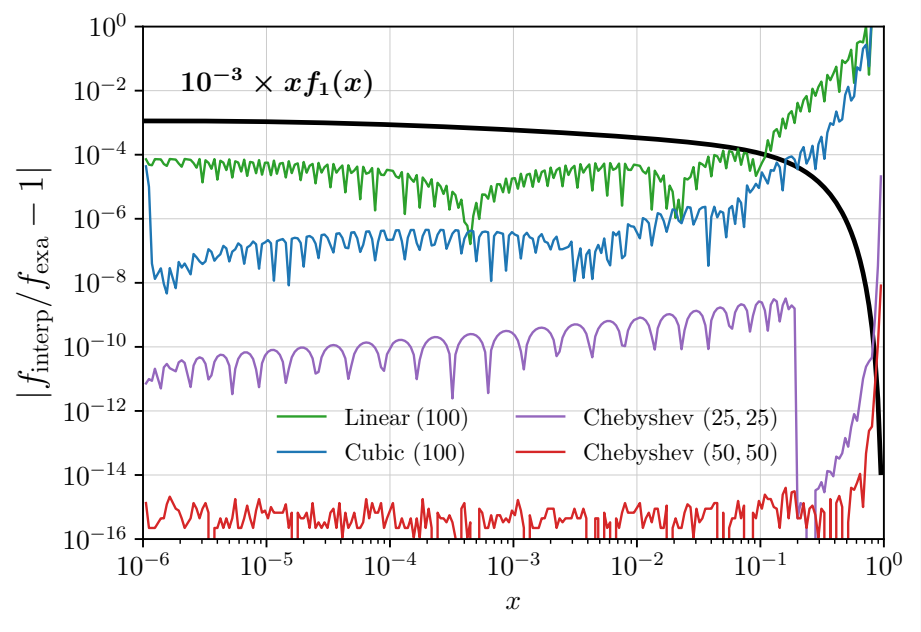
Key advantages:

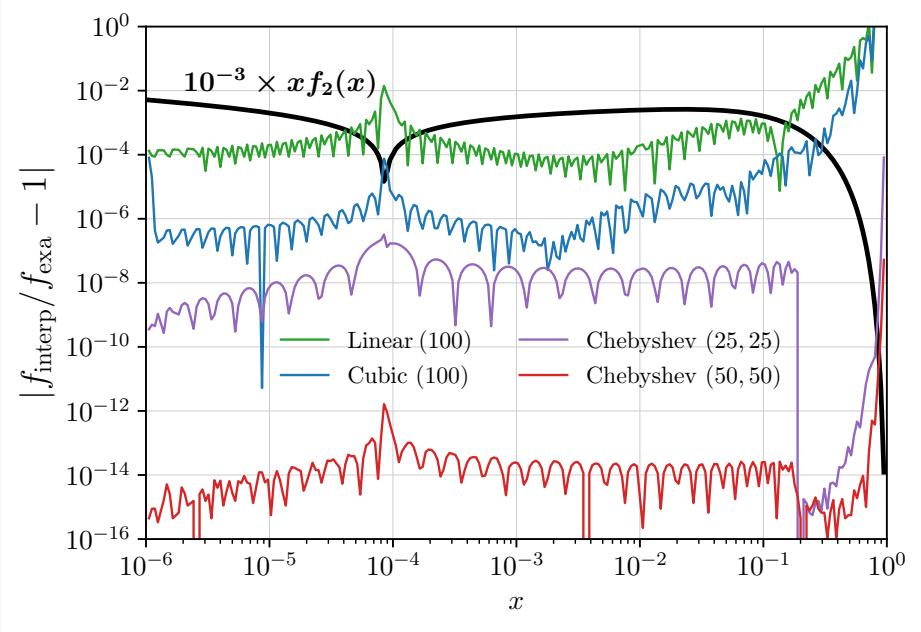
- C^∞ continuous $\Longleftrightarrow$ yields numerically stable derivations up to very high order
- For smooth (analytical) functions, the error decreases **exponentially** with the number of grid points ($\neq$ polynomial dreacrease in piecewise interpolations such as cubic spline)
- Chebyshev grid points are defined on **Chebyshev nodes** $\Longleftrightarrow$ requires fewer nodes

To benchmark the accuracy & precision of the Chebyshev interpolation, we consider two representative test-functions:

$$xf_1(x) = 0.0703x^{-0.415(1+4.44x)(1+0.0373\ln x)}(1-x)^{7.75} \quad \text{(ABMP16 parametrisation of } f_{\overline{u}})$$

$$xf_2(x) = 4.34x^{-0.015}(1-x)^{9.11} - 1.048x^{-0.167}(1-x)^{25.0} \quad \text{(HERAPDF20 parametrisation of } f_g)$$

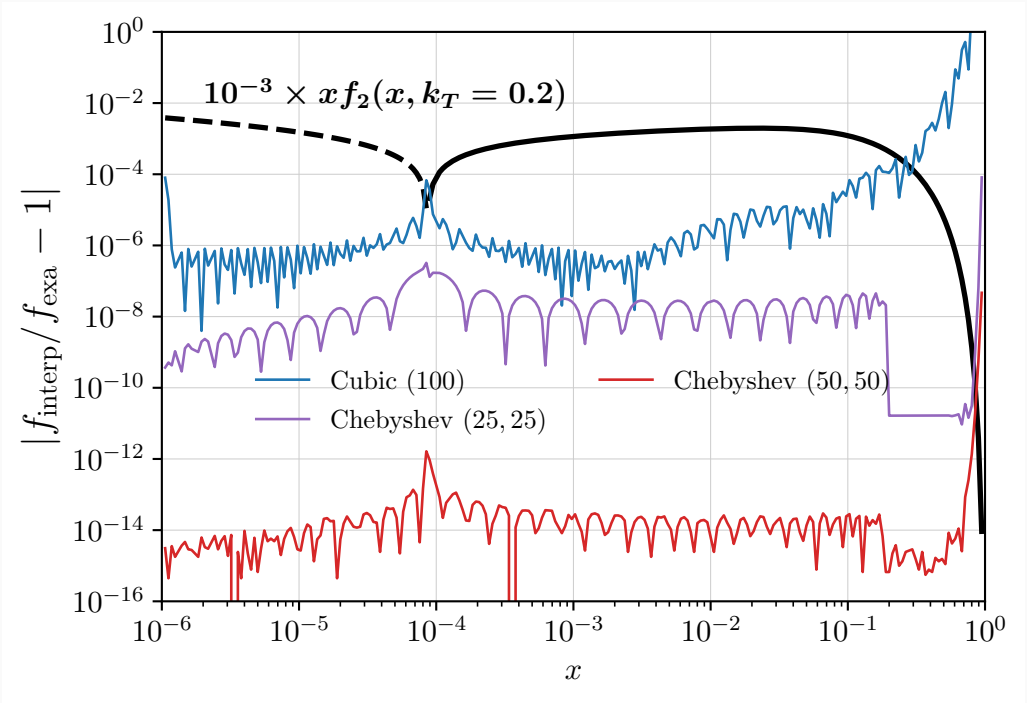


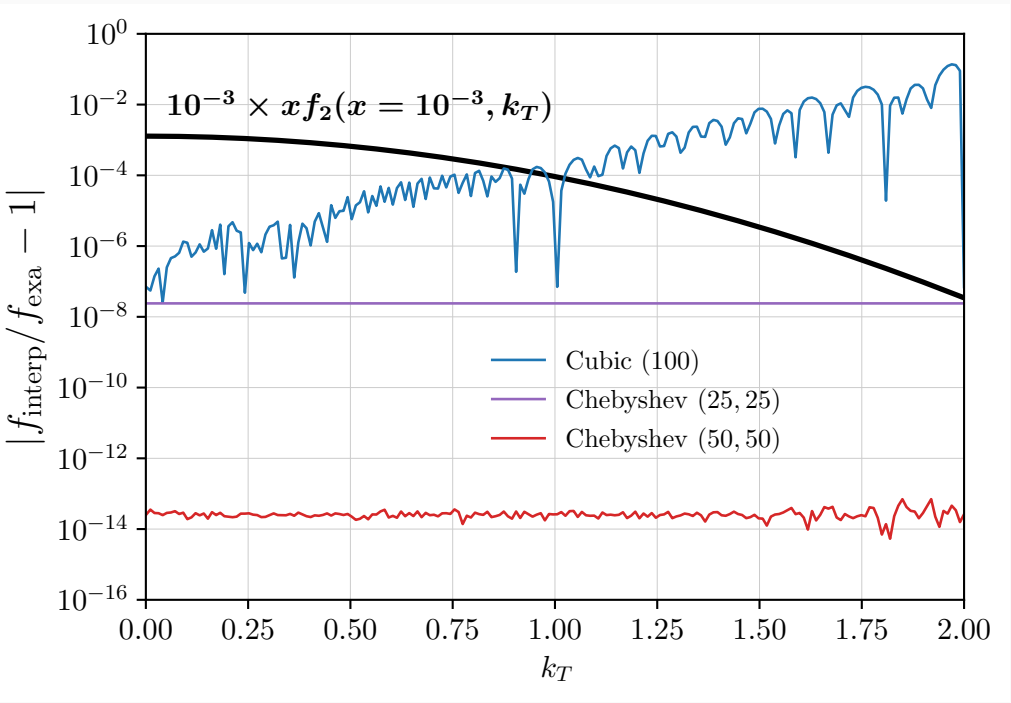


Using Chebyshev Grid $\Longrightarrow$ Another factor of 2 **reduction** in "file size"

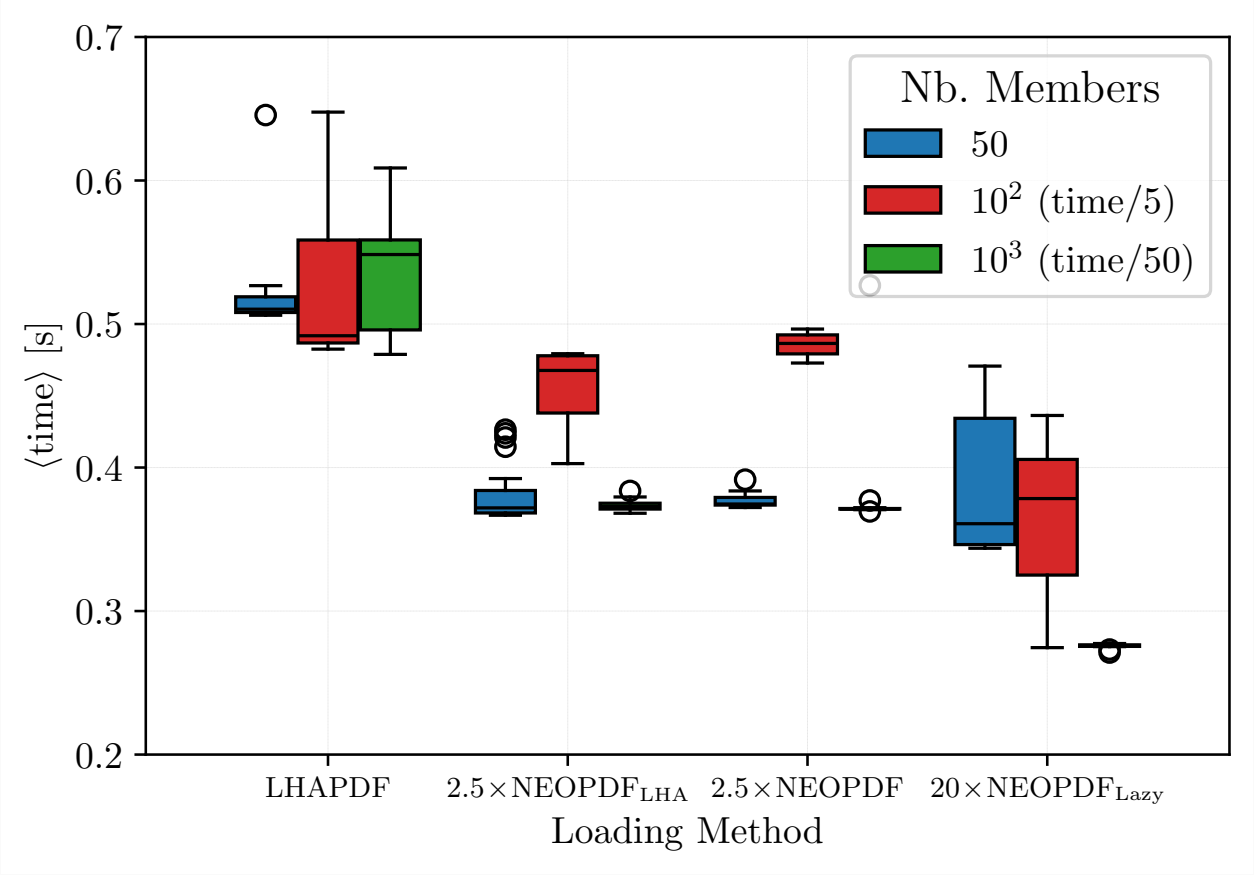
Use the same parametrisation as for the collinear PDFs and add k_T^2-dependence using **"Gaussian Ansatz"**:

$$xf_i(x, k_T^2) = xf_{i(x)} \frac{1}{\pi \langle k_T^2(x) \rangle} \exp\left(-\frac{k_T^2}{\langle k_T^2(x) \rangle} \right)$$





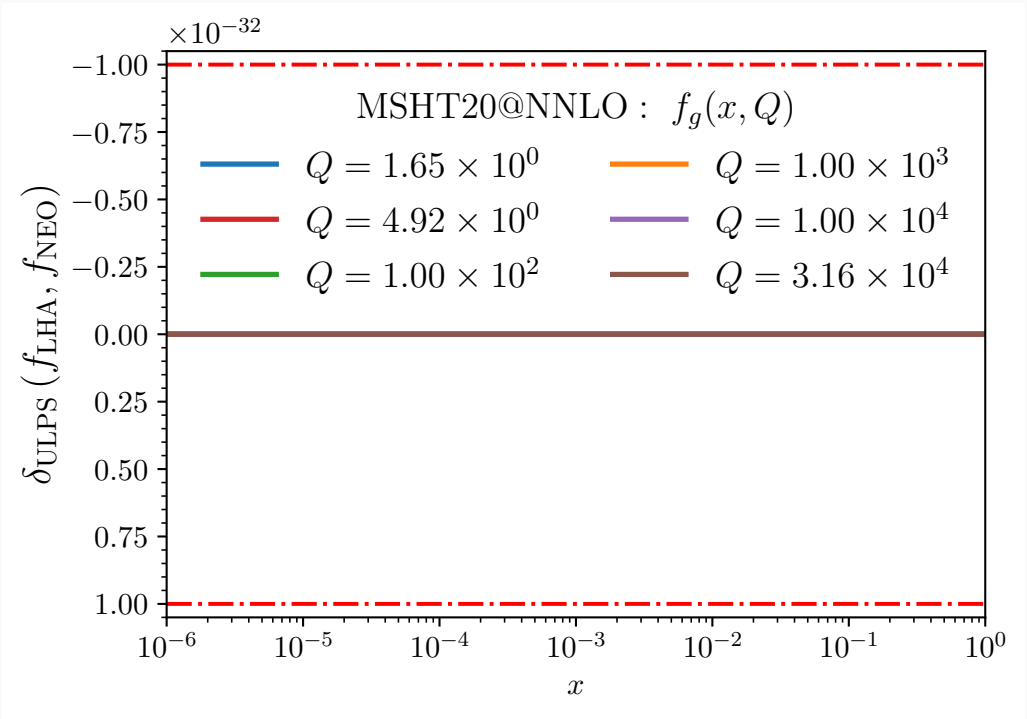
NeoPDF (with its different loading methods) is at least 2.5 times faster in loading PDF sets with multiple members

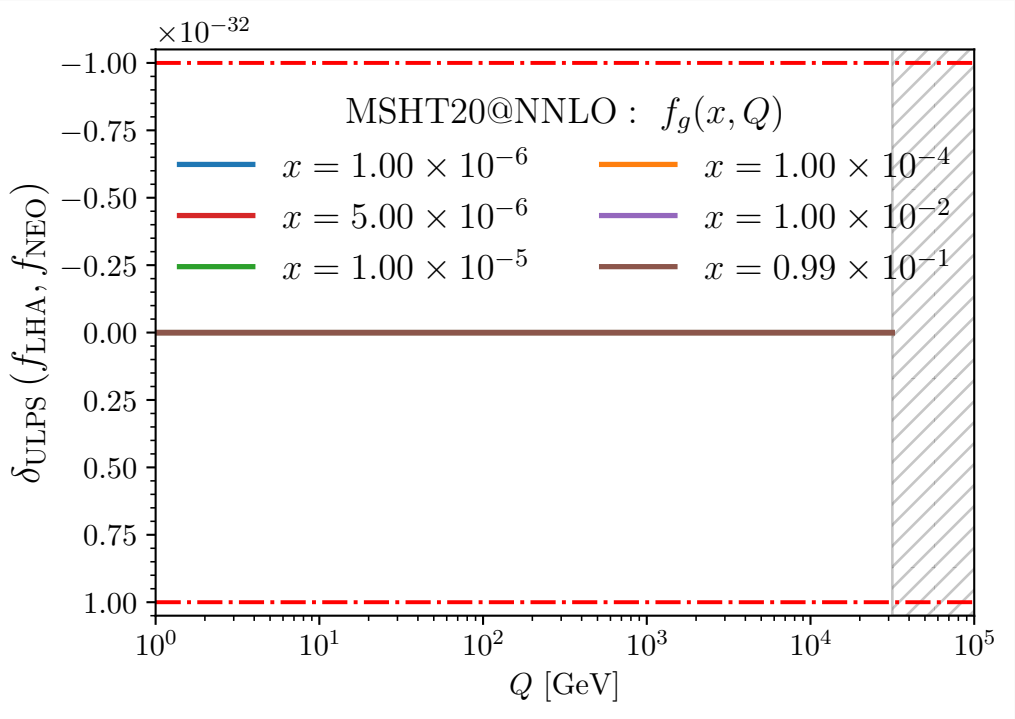


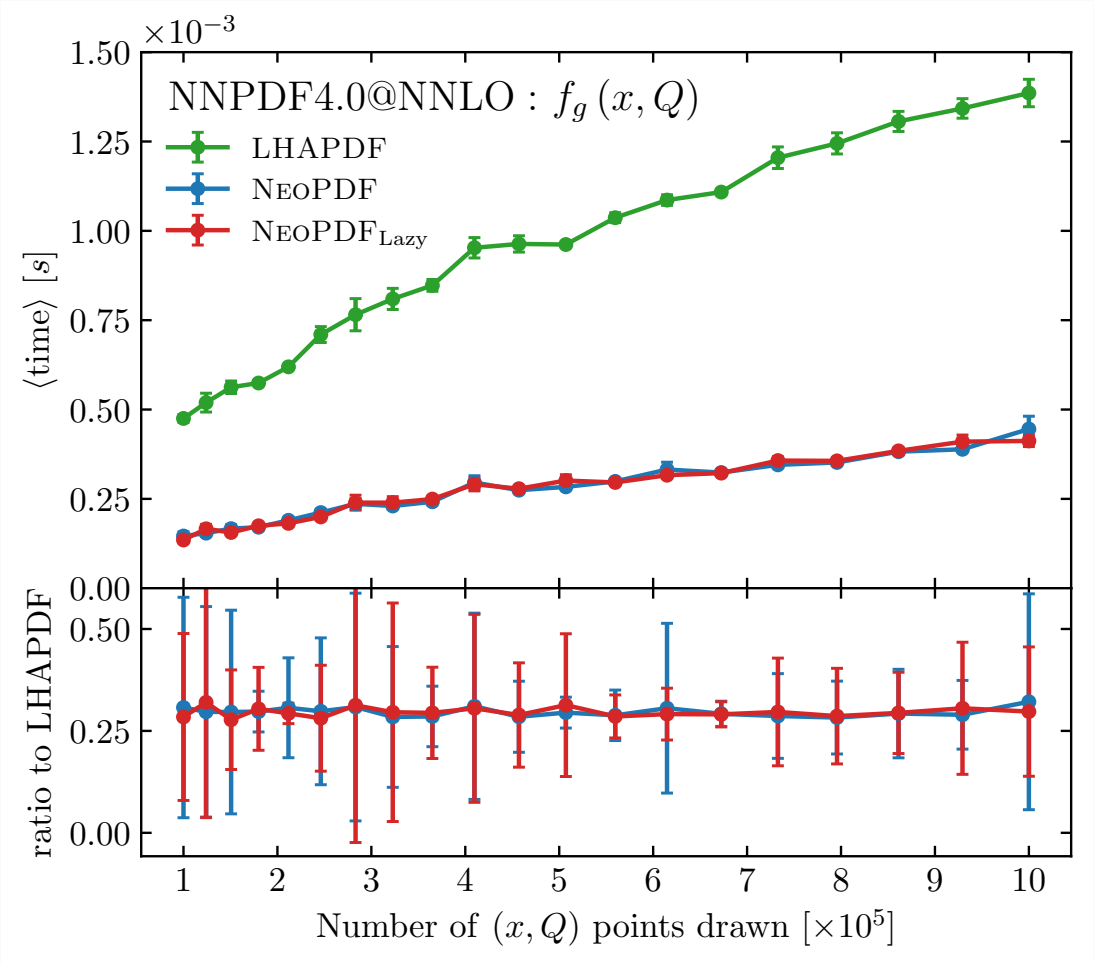
We consider the "Difference in Units of Last Place" δ_{ULPS} which given two floating-point numbers $f_a, f_b \in \mathbb{F}_{64}$ is defined as:

$$\delta_{\text{ULPS}} = |\text{ULP}\ (f_a) - \text{ULP}\ (f_b)|, \quad \text{ULP} : \mathbb{F}_{64} \longmapsto \mathbb{Z} \quad \text{(in IEEE-754 encodings)}$$

A discrepancy of **one**-δ_{ULPS} correspond to a relative error of $\mathcal{O}(10^{-16})$, while $\delta_{\text{ULPS}} = 0$ means exact equality in binary repr.

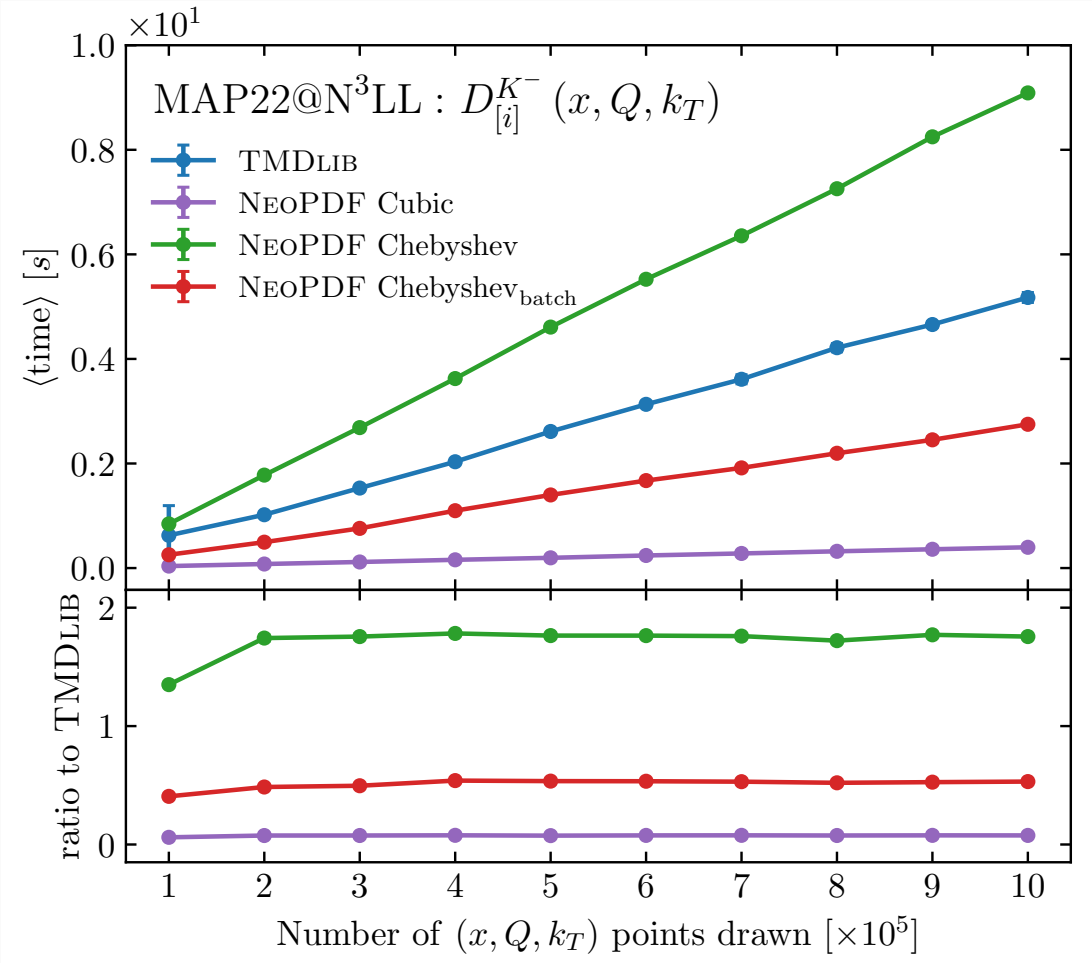






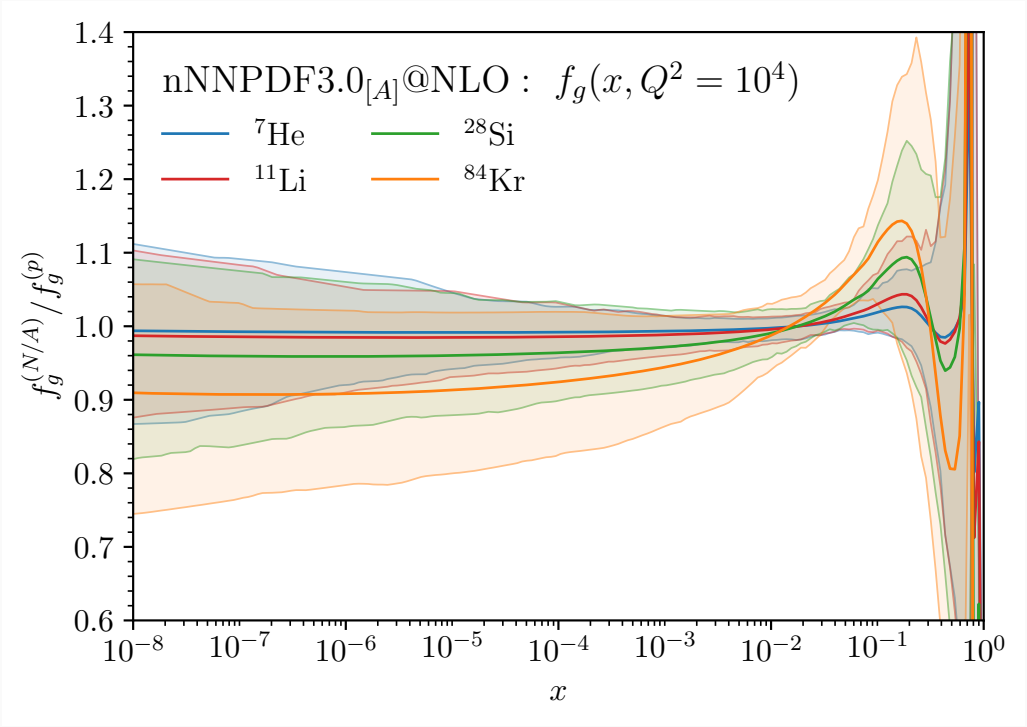
NeoPDF achieves a substantial performance improvement over LHAPDF; the average interpolation time is reduced by at least a factor of 2.

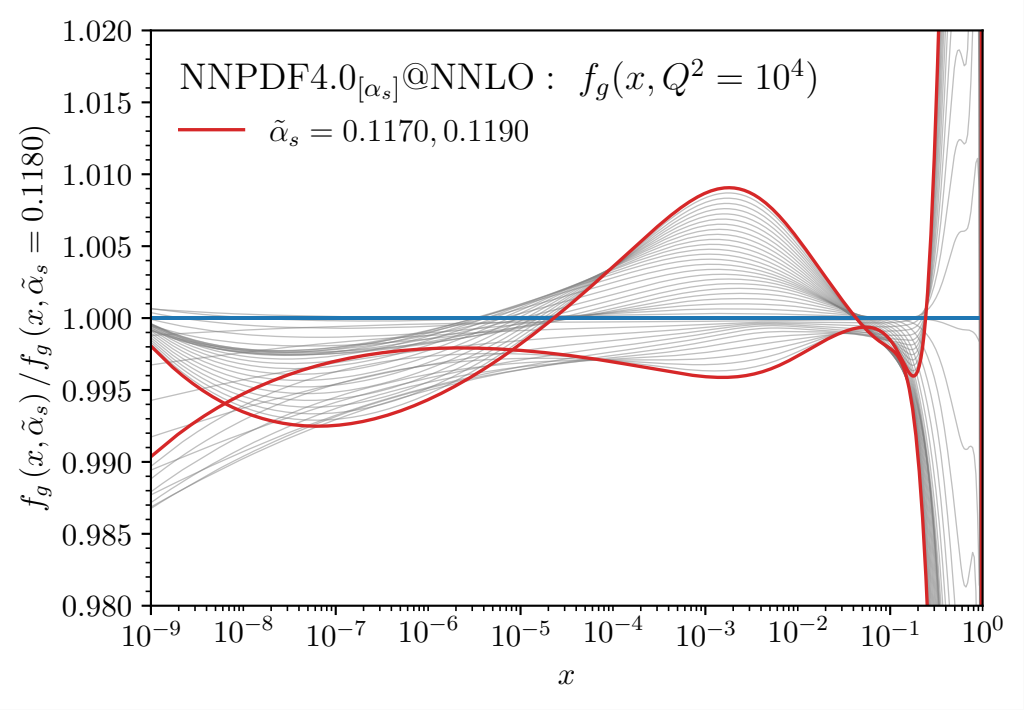
$\Longrightarrow$ Such a reduction directly translates into practical benefits where PDF evaluations can dominate the runtime of Monte Carlo simulations.



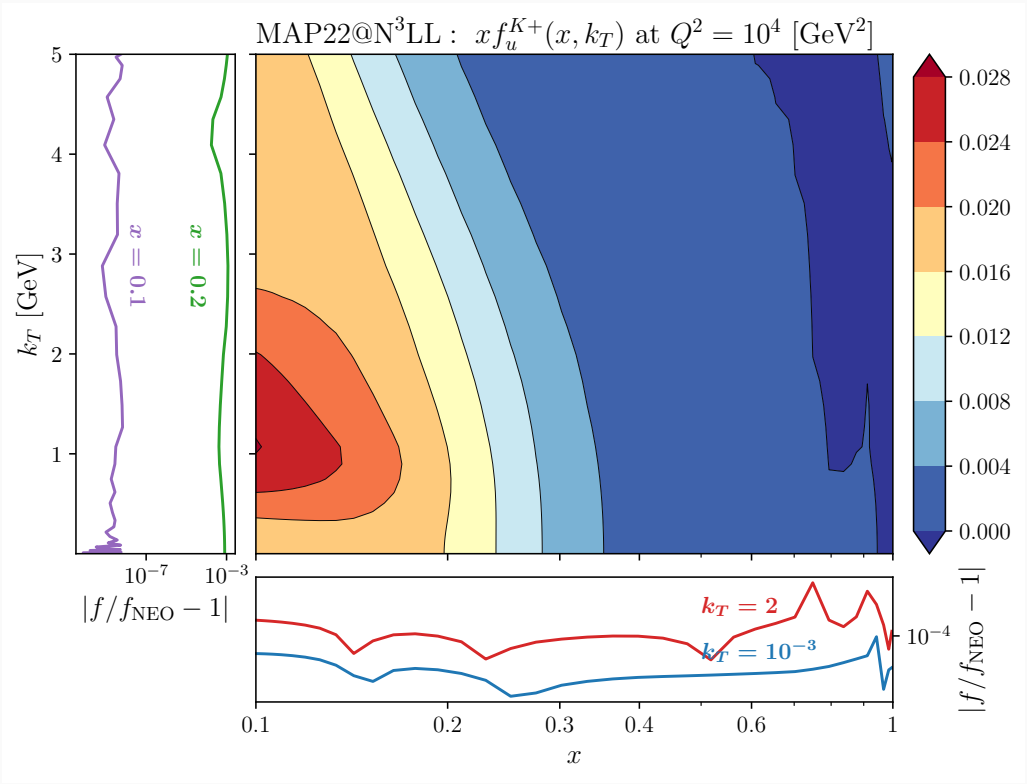
NeoPDF achieves a substantial performance improvement over TMDlib; the average interpolation time is reduced by at least a factor of 5 when using the same interpolation.

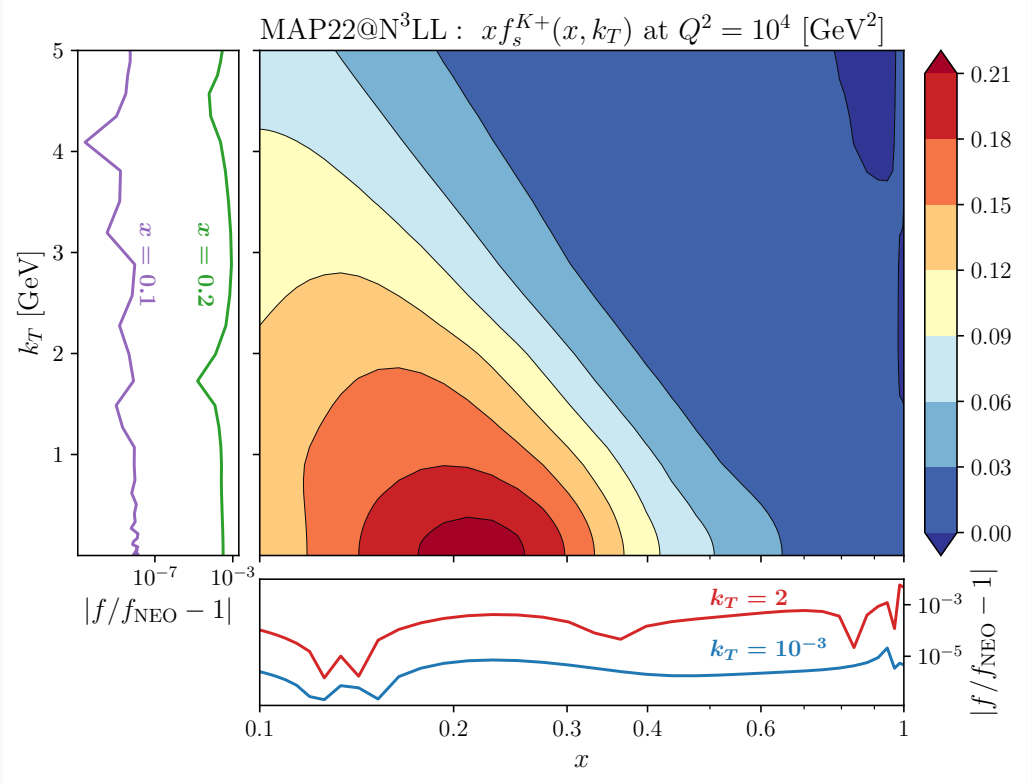
NeoPDF **consistently** supports interpolations along the atomic mass number A (for nuclear PDF fits & studies) and the strong coupling $\tilde{\alpha}_s = \alpha_s(M_Z)$ (for various phenomenological studies).





NeoPDF inherently supports interpolation along the parton's transverse momentum, making it suitable for TMD PDFs





Documentation: https://qcdlab.github.io/neopdf/

NeoPDF maintains API compatibility with LHAPDF to enable seamless migration: existing LHAPDF codes can be migrated by simply changing the *import* statements; function signatures are preserved (e.g. *mkPDF()*, *mkPDFs()*, *xfxQ2()*, *alphasQ2()*).

The Python API can be installed with any Python Package Manager:

```
> pip install neopdf-hep
```

Example of how NeoPDF is called:

```python
from neopdf.pdf import PDF as lhapdf

pdf = lhapdf.mkPDF("NNPDF40_nnlo_as_01180") # To use the NeoPDF format, append PDF name with `.neopdf.lz4`

xf = pdf.xfxQ2(21, 1e-3, 1e4)
alphas_q2 = pdf.alphasQ2(1e4)

xfs = pdf.xfxQ2s(pids=[1, 2, 3], xs=[1e-3, 1e-2], q2s=[1e2, 1e4]) # supports multi-threading out of the box for Parallelization
```

Note: The "No-Code Migration" is **not fully supported** with the Fortran, C, and C++ interface.

NeoPDF provides a Command Line Interface (CLI) tool, allowing users to perform PDF operations/interpolations, file format conversion, sets combination, metadata inspection, etc. directly from the terminal.

The NeoPDF CLI can also be installed easily using any Python Package Manager:

```
> pip install neopdf-cli
```

To compute PDF interpolations:

```
> neopdf compute xfx_q2 --pdf-name NNPDF40_nnlo_as_01180 --member 0 --pid 21 1e-3 1e4
```

To convert a set in the LHAPDF format into NeoPDF:

```
> neopdf write convert --pdf-name NNPDF40_nnlo_as_01180 --output NNPDF40_nnlo_as_01180.neopdf.lz4
```

To combine multiple nuclear PDF sets into a single NeoPDF file in order to make the dependence on A explicit:

```
> neopdf write combine-npdfs --pdf-names <list_pdf_names> --output nNNPDF30_nlo_as_0118.neopdf.lz4
# Similar syntax exist to combine multiple sets with differrent `alphas` values.
```

Conclusions & Outlook

PineAPPL:

- PineAPPL provides a framework to efficiently manipulate generic processes in QCD
- PineAPPL supports an arbitrary number of convolutions with a consistent treatment of electroweak corrections & fragmentation scales

NeoPDF:

- NeoPDF is already at an advanced stage: fully compatible with LHAPDF and can do more (it outperforms LHAPDF in many aspects)
- NeoPDF supports **generic** non-perturbative distribution functions, including TMDs, GPDs, and GTMDs
- NeoPDF introduces a more accurate and efficient "**Chebyshev Interpolation**": more precise and numerically stable; requires much lower number of grid points
- Support for FONLL will come soon

References

[Adam and others, 2018] Adam, J., and others. (2018). Longitudinal Double-Spin Asymmetries for $pi^{\{0\}}$s in the Forward Direction for 510 GeV Polarized pp Collisions. *Phys. Rev. D*, *98*(3), 32013.

[Airapetian and others, 2019] Airapetian, A., and others. (2019). Longitudinal double-spin asymmetries in semi-inclusive deep-inelastic scattering of electrons and positrons by protons and deuterons. *Phys. Rev. D*, *99*(11), 112001.

[Ball et al., 2016] Ball, R. D., Nocera, E. R., and Rojo, J. (2016). The asymptotic behaviour of parton distributions at small and large x. *Eur. Phys. J. C*, *76*(7), 383.

[Bertone, 2018] Bertone, V. (2018). APFEL++: A new PDF evolution library in C++. *Pos*, 201.

[Bertone et al., 2014] Bertone, V., Carrazza, S., and Rojo, J. (2014). APFEL: A PDF Evolution Library with QED corrections. *Comput. Phys. Commun.*, *185*, 1647–1668.

[Boughezal et al., 2021] Boughezal, R., Li, H. T., and Petriello, F. (2021). W-boson production in polarized proton-proton collisions at RHIC through next-to-next-to-leading order in perturbative QCD. *Phys. Lett. B*, *817*, 136333.

[Candido et al., 2024] Candido, A., Hekhorn, F., Magni, G., Rabemananjara, T. R., and Stegeman, R. (2024). Yadism: yet another deep-inelastic scattering module. *Eur. Phys. J. C*, *84*(7), 698.

[Carrazza et al., 2020] Carrazza, S., Nocera, E. R., Schwan, C., and Zaro, M. (2020). PineAPPL: combining EW and QCD corrections for fast evaluation of LHC processes. *JHEP*, *12*, 108.

[Cruz-Martinez et al., 2025] Cruz-Martinez, J., Huss, A., and Schwan, C. (2025). Fast interpolation grids for the Drell–Yan process. *Eur. Phys. J. C*, *85*(4), 459.

References

[Devoto et al., 2025] Devoto, S., Jezo, T., Kallweit, S., and Schwan, C. (2025). MATRIX HAWAII: PineAPPL interpolation grids with MATRIX.

[Diehl et al., 2022] Diehl, M., Nagar, R., and Tackmann, F. J. (2022). ChiliPDF: Chebyshev interpolation for parton distributions. *Eur. Phys. J. C*, *82*(3), 257.

[Dulat et al., 2018] Dulat, F., Mistlberger, B., and Pelloni, A. (2018). Differential Higgs production at N^{3}LO beyond threshold. *JHEP*, *1*, 145.

[Grazzini et al., 2018] Grazzini, M., Kallweit, S., and Wiesemann, M. (2018). Fully differential NNLO computations with MATRIX. *Eur. Phys. J. C*, *78*(7), 537.

[Huss and others, 2025] Huss, A., and others. (2025). NNLOJET: a parton-level event generator for jet cross sections at NNLO QCD accuracy.

[Jager et al., 2003] Jager, B., Schafer, A., Stratmann, M., and Vogelsang, W. (2003). Next-to-leading order QCD corrections to high p(T) pion production in longitudinally polarized pp collisions. *Phys. Rev. D*, *67*, 54005.