# Pineline: Industrialization of High-Energy Theory Predictions

Andrea Barontini[a], Alessandro Candido[a], Juan M. Cruz-Martinez[b,*], Felix Hekhorn[a], Christopher Schwan[c]

[a]*TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano and INFN Sezione di Milano,*
*Via Celoria 16, 20133, Milano, Italy*
[b]*CERN, Theoretical Physics Department, CH-1211 Geneva 23, Switzerland*
[c]*Universität Würzburg, Institut für Theoretische Physik und Astrophysik, 97074 Würzburg, Germany*

## Abstract

We present a collection of tools efficiently automating the computation of large sets of theory predictions for high-energy physics. Calculating predictions for different processes often require dedicated programs. These programs, however, accept inputs and produce outputs that are usually very different from each other. The industrialization of theory predictions is achieved by a framework which harmonizes inputs (runcard, parameter settings), standardizes outputs (in the form of grids), produces reusable intermediate objects, and carefully tracks all meta data required to reproduce the computation. Parameter searches and fitting of non-perturbative objects are exemplary use cases that require a full or partial re-computation of theory predictions and will thus benefit of such a toolset. As an example application we present a study of the impact of replacing NNLO QCD K-factors with the exact NNLO predictions in a PDF fit.

*Keywords:* Parton distributions, grids, reproducibility

## PROGRAM SUMMARY

*Program Title:* `pineline`

*Program URL:* https://nnpdf.github.io/pineline/

*Licensing provisions:* GPLv3

*Programming language:* Python, Rust

*Nature of the problem:* The computation of theoretical quantities in particle physics often involves computationally-intensive tasks such as the calculation of differential cross sections in a systematic and reproducible way. Different groups often use different conventions and choices which makes tasks such as the fitting of physical parameters or quantities very computationally challenging and hard to reliably reproduce.

*Solution method:* We create a pipeline of tools such that a user can define an observable and a theory framework and obtain a final object, containing all relevant theoretical information. Such objects can be then used in a variety of interchangeable ways (fitting, analysis, experimental comparisons).

## 1. Introduction and motivation

Modern particle physics phenomenology is increasingly reliant on complex theoretical calculations whose accuracy needs to match very precise measurements, chiefly the ones from experiments at the Large Hadron Collider (LHC) [1]. An increase in accuracy in those predictions is associated to the computation of higher orders in the strong and/or electroweak couplings for partonic cross sections, and usually performed by numerical programs, which we will call *generators* throughout this paper. Since the computations are very demanding in runtime, memory and storage, these generators are usually optimized for and can only calculate a small set of observables, and, furthermore, they often use different conventions and strategies. Being able to generate, store and exchange predictions in well-suited formats for a large set of processes, such that they can be utilized for a variety of analyses, is therefore advantageous.

In this paper we propose a framework, which we call `pineline`, that aims to generate theory predictions by 1) building a translation layer from a common input format to each of the different generators and 2) implementing a common output format for all of them. This is the idea that we call *industrialization*: while specific generators are sufficient for the calculation of single processes, there is no single generator that is able to calculate all processes, which are not necessarily limited to processes at the LHC, but may also include deep-inelastic scattering processes, for example. By interfacing to multiple generators, and thus connecting them in an "assembly line" or "pipeline", we can easily run the generator best suited for a particular process, and by having a common input format we can

easily perform variations, such as changing parameters for parameter scans.

The motivation for this project is fitting parton distribution functions (PDFs) [2–5], but the output generated by `pineline` can potentially be used in any fit or analysis that requires theory predictions. One interesting feature of a PDF fit in this context is that a very large number of predictions go into it. This complicates keeping track of the theory parameters used, for example. While this is a manageable problem for a few predictions, for a complete PDF fit it is crucial to make sure that different processes make use of sets of parameters that are compatible with each other. Keeping track of the parameters in a central place makes it then easily possible to be able to rerun predictions if we want to change (some) of those parameters, for example. We stress that PDFs are a fundamental ingredient in any observable involving hadrons in the initial state and thus should be controlled in all applications.

The rest of the paper is organised as follows: in Section 2 we outline the abstract ideas that led us in the design of the `pineline`. In Section 3 we give a brief technical overview of the actual implementation leaving the details to the respective documentations. In Section 4 we give an explicit application of the framework, before concluding in Section 5.

## 2. Guiding principles

We aim to bring theory predictions in high-energy physics closer to the FAIR principles [6] (findability, accessibility, interoperability, and reusability) to allow for sustainable and reproducible research.

### 2.1. Input and output formats

Our framework is built to generate and store theory predictions in a single format from a common set of inputs. By making the input common across different generators we can enforce consistency in theory settings, and, by storing them in a single format, we ensure they can be used and analyzed regardless of how they were computed originally.

To give an impression of the diversity of generators, in NNPDF4.0 [2] the predictions from more than ten different programs were used: `APFEL` [7], `DYNNLO` [8, 9], `FEWZ` [10–12], `Madgraph5_aMC@NLO` [13, 14], `MCFM` [15–18], `Njetti` [19, 20], `NNLOjet` [21], `NLOjet++` [22], `Top++` [23], `Vrap` [24] and `SHERPA` [25]. Each of these programs requires a different set of inputs and parameters to run, and even when they are similar they are provided in different formats. To mitigate this problem we propose a layout with a global "theory runcard" which, through an appropriate generator-dependent translation layer, is fed to the target program.

The output of the programs is a hadronic observable, which means it has already been folded with non-perturbative objects, such as the PDF. By standardizing the output of all generators to be an interpolation grid we can reanalyze

the same prediction in different scenarios, without requiring an (expensive) recomputation. The evaluation of the results for different sets of PDFs becomes almost instantaneous. As a by-product, it also facilitates parameter fits for objects that depend on those quantities.

In the context of PDF fitting we can think of two common scenarios:

- the inclusion of new data points into the fit (coming from existing or new experiments [26–28])

- investigate the impact of theory settings (such as the reference value of the strong coupling $\alpha_s(M_Z^2)$ [29]).

Both require us to (re-)compute theory predictions for a large number of data points. To give a concrete example of the scale of the problem, let us consider again NNPDF4.0 which fits more than 4500 data points across almost 100 different datasets. In order to match the increasing demands from the theory side we require more and more automation to avoid time-consuming and error-prone manual processes.

The actual objects we are working with in practice are interpolation grids [30–32], which store theory predictions independently of PDFs and the strong coupling. Interfaces of them to some generators are available [33–35]. Being independent of PDFs they are ideally suited for PDF fits where they have been widely adopted, but their use is not limited to this area. Note that by re-fitting the PDF, any observable that depends on it will change. However, the partonic cross sections do not depend on the PDFs. By having them stored as interpolation grids, one can update all predictions without recomputing the most computationally heavy part of the observables.

In summary, our goal is to provide a reliable and easy-to-use workflow that connects the necessary intermediate steps and that can be scaled to any amount of data.

### 2.2. Reproducibility

A very important aspect of joining all of these different generators in a pipeline is the reproducibility of the results: it must always be possible to trace every prediction back to its inputs, so that any result can be independently checked by a third-party, and so that the impact of the change from a base set of parameters can be calculated. To this end, each interpolation grid and all intermediate objects contain all the (meta)data needed to recalculate them and to verify that both are compatible with each other. In particular, this includes: the programs used, their version numbers and random seeds, the value of relevant standard model parameters, renormalization scheme choices, phase space cuts, and Monte Carlo uncertainties. We note that many interpolation grids publicly available on hepdata [36] and ploughshare [37] do not include this information, though sometimes it can be inferred from the associated publications. However, often these data are not available, making comparisons more difficult and time-consuming. We make that metadata explicitly available

in the grids and all other outputs, from which it can be reliably and easily extracted.

## 2.3. Open-source software

All the software used in this framework is open source, to facilitate its distribution, use and maintenance. In addition to the code, also the data are available online in formats that can be analysed with open-source tools. Specifically, we store all metadata in the widely used YAML[1] format while interpolation grids are stored as PineAPPL grids, which can be interfaced to with many programming languages.

Finally, we note that this work can also be seen as a continuation of the effort already started with the publication of the NNPDF fitting code [38], giving the community all necessary tools to reproduce and perform (theory) variations of NNPDF fits.

## 3. Technical overview

In the following we describe the technical implementation of the ideas highlighted above into the `pineline`. In order to do so, it is easiest to follow the *deliverables*, i.e. the objects that the `pineline` produces. These are shown in Fig. 1 and are the oval objects, namely 1) PineAPPL grids, 2) evolution-kernel operators (EKOs) and 3) fast-kernel (FK) tables. PineAPPL grids, like APPLgrids and fastNLO tables, store theoretical predictions independently from their PDFs and the strong coupling. EKOs and FK tables are tailored towards PDF fits, and translate interpolation grids to use a single factorization scale.

An extended discussion of the technical details of the various programs is beyond the scope of this paper. We refer the interested reader instead to the relevant documentation and development repositories of each tool.

## 3.1. Mathematical overview

Let us consider the calculation of a single observable $\sigma$, which for the sake of readability we assume to contain only a single convolution, e.g., for the case of a DIS structure function. The extension to more convolutions is straightforward. Eq. (1) shows the defining property of interpolation grids, namely how convolutions with PDFs $f_a(x, \mu_F^2)$ are performed:

$$\sigma = \sum_{i,j,k} \sum_a f_a(x_i, \mu_{Fj}^2) \alpha_{\rm s}^{n+k}(\mu_{Rj}^2) \sigma_a^{(k)}(x_i, \mu_{Fj}^2, \mu_{Rj}^2). \quad (1)$$

The grid itself is the set of values $\left\{\sigma_a^{(k)}(x_i, \mu_{Fj}^2, \mu_{Rj}^2)\right\}$ for all partons $a$ and perturbative orders $k$. Note that the PDFs are interpolated, and therefore evaluated at specific momentum fractions $\{x_i\}$ and (squared) factorization scales $\{\mu_{Fj}^2\}$, just as the partonic cross sections $\sigma_a$. For simplicity, in what follows we take the renormalization

scale to be equal to the factorization scale $\mu = \mu_{\rm R}^2 = \mu_{\rm F}^2$, but the choice of scale is completely free.

The interpolation transforms the convolution integral to a sum, resulting in the grid being a PDF-independent quantity. In particular, the PDF is expanded over an interpolation basis, with the expansion coefficients being the values of the PDF on some nodes. This means the specific interpolation basis is only used in the construction of the grid, but is not relevant for the construction of the PDF table (and so not of concern for any PDF user).

To represent interpolation grids we use the PineAPPL library [35]. The source code can be inspected from its repository

<div align="center">

`https://github.com/NNPDF/pineappl`

</div>

and the associated documentation consulted at

<div align="center">

`https://nnpdf.github.io/pineappl/`.

</div>

For the special case of PDF fits, interpolation grids are not the most efficient representation yet, given that the factorization dependence of the PDFs is known perturbatively and consequently not fitted. We can therefore rewrite Eq. (1) to refer only to a single factorization scale $\mu_0$, which in PDF fits is known as the initial scale or the fitting scale:

$$\sigma = \sum_i \sum_a f_a(x_i; \mu_0^2) \, {\rm FK}_a(x_i; \mu_0^2). \quad (2)$$

The object $\{{\rm FK}_a(x_i; \mu_0^2)\}$ is known as a fast-kernel (FK) table [39] and is a special case of an interpolation grid that

- uses a single factorization scale and

- contains the resummed evolution, thus combining various perturbative orders and therefore consuming the dependence on the strong coupling.

An FK table can be computed using evolution kernel operators (EKOs),

$$\mathrm{FK}_a(x_i; \mu_0^2) = \sum_{b,j,k,l} \alpha_{\rm s}^{n+k}(\mu_j^2) \, \mathrm{EKO}_{a,i}^{b,l,j} \, \sigma_b^{(k)}(x_l, \mu_j^2), \quad (3)$$

where $\mathrm{EKO}_{a,i}^{b,l,j}$ are the (linear) operators resulting from the evolution equations. FK tables are ideally suited for PDF fits, because the time- and memory-consuming evolutions are done only once and not during the fit.

What we have gained are theoretical predictions $\{\sigma\}$, represented as FK tables, which allow us to perform convolutions with a set of one-dimensional PDFs $f_a(x; \mu_0^2)$ very efficiently. However, the price we have to pay is that we need a set of tools that calculate all the required objects:

1. A numerical calculation must generate interpolation grids for each observable $\sigma$ that we want to incorporate in a fit.

---

2. Next, we need to calculate the EKOs, for the corresponding choices in each observable calculated previously and the choices made in the fit.

3. Finally, we need to evolve the interpolation grids using the EKOs to generate FK tables.

In the subsequent sections we briefly review the various programs dedicated to each step.

Note that the assumption of a single scale is chosen here only to simplify the notation, but this is not present in the actual implementation. In fact, having chosen a modularized composition of the `pineline` allows for a simplified implementation of scale variations: scale variation can be divided into renormalization scale variation, related to the ultraviolet structure of the partonic matrix elements and which can thus only act on the level of grids, and factorization scale variation related to the collinear factorization theorem, which can either effect the split between PDFs and grids or directly EKOs. We can use such scale variations to estimate the uncertainty associated to the limited perturbative knowledge of perturbative QCD [40].

### 3.2. Generating grids: `pinefarm`

PineAPPL itself is physics agnostic and therefore we need a parton-level generator to create and actually fill the grids. This requires a generator to be interfaced to PineAPPL, which then sends the relevant phase-space information, i.e. $x, \mu_F, a, \ldots$, to PineAPPL, which collects it in a space-efficient data structre representing $\{\sigma_a^{(k)}(x_i, \mu_j^2)\}$ (see Eq. (1)). Practically, this is done using an interface offered by PineAPPL, available for the programming languages C, C++, Fortran, Python and Rust.

As of now, PineAPPL has been interfaced to the following generators:

- Madgraph5_aMC@NLO [13, 14] to calculate LHC processes, including NLO EW and QCD–EW corrections,

- yadism [41] to calculate NC and CC DIS processes,

- a modified version[2] of Vrap [24] for fixed-target Drell–Yan processes, and

- an interface to MATRIX [42] is in progress.

Furthermore, `PineAPPL` can convert already existing APPLgrids and fastNLO tables into its own format using its command-line interface (CLI). See App. Appendix A.1 for an example.

The program `pinefarm`, presented here for the first time, abstracts away most of the differences of different generators. For the generators listed above it recognizes different input files, which specify the requested physical observable. It also performs substitutions from a theory parameters database, and directly runs the generators to produce predictions and collect the desired interpolation grid. The extension to more generators should be straightforward thanks to the open source nature of `PineAPPL` and `pinefarm`.

The source code can be inspected from its repository

https://github.com/NNPDF/pinefarm

and the associated documentation inspected at

https://pinefarm.readthedocs.io.

### 3.3. Generating evolution kernel operators: `EKO`

While grids $\{\sigma_a^{(k)}(x_i, \mu_j^2)\}$ are convoluted with PDFs evaluated at high scales $\mu_j^2$, FK tables $\{FK_a(x_i; \mu_0^2)\}$ are convoluted with PDFs evaluated at the fitting scale $\mu_0^2$ reducing the dimensionality thus to just two dimensions for DIS observables (parton flavor index and momentum fraction) and four for hadronic observables. This reduction is possible because the scale dependence of PDFs is given by the DGLAP equation [43–45].

The software package `EKO` [46, 47] has been developed to solve these equations in terms of evolution kernel operators (EKOs):

$$f_b(x_l, \mu_j^2) = \sum_i \sum_a \mathrm{EKO}_{a,i}^{b,l,j} \, f_a(x_i; \mu_0^2) \qquad (4)$$

In contrast to similar programs [7, 48–50] `EKO` focuses specifically on the direct computation of the operator which allows the described pipeline to use them to produce FK tables. Since the operator itself is PDF independent it allows also to reuse existing operators just like reusable tools in the theory factory.

The source code can be inspected from its repository

https://github.com/NNPDF/eko

and the associated documentation is available at

https://eko.readthedocs.io.

### 3.4. Generating FK tables: `pineko`

Interpolation grids and EKOs are joined together in `pineko`, presented here for the first time, to produce FK tables according to Eq. (3). Specifically, `pineko` has to extract the relevant information from a grid and a theory runcard (containing all the relevant theory parameters) and then pick or, if it has not been calculated yet, compute the required EKO as described in Section 3.3. Once the EKO is computed, `pineko` loads the grid and evolves it using the EKO to produce the final FK table.

Since Eq. (2) is a special case of Eq. (1), PineAPPL can also represent FK tables in the same format. This serves an important purpose: at any point in the pipeline, a theory prediction, whether it is an interpolation grid or an FK table, whether it was created using a Monte Carlo generator or converted from other interpolation grids, is

---
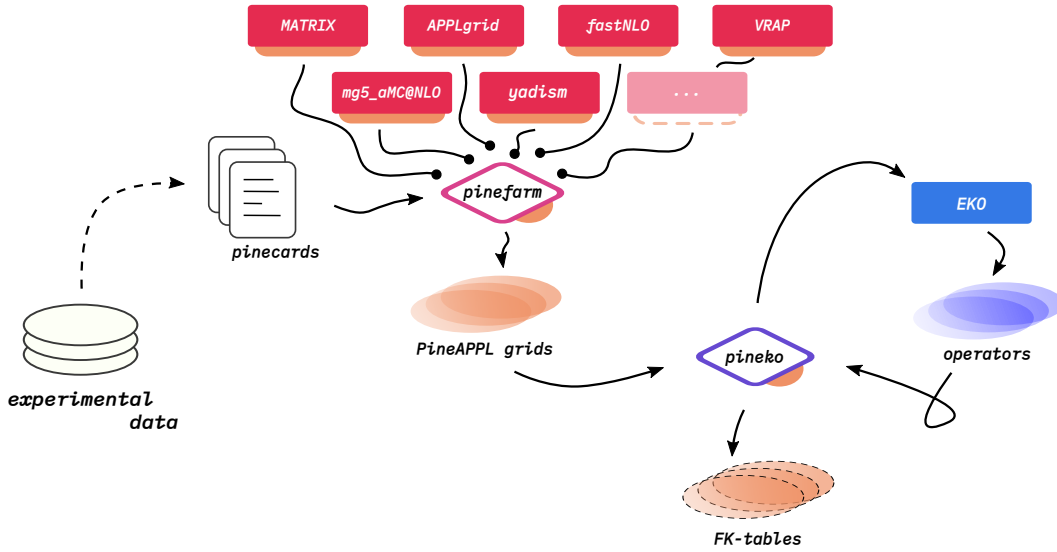
[2]https://github.com/NNPDF/hawaiian_vrap

Figure 1: Flow diagram showing the overall pipeline architecture and deliverables in the case of parameter fits. Arrows in the picture indicate the flow of information (together with the execution order) and the orange insets on other elements indicate an interface to PineAPPL. The programs `pinefarm` and `pineko` act as interfaces between other programs and the deliverable objects, represented by ovals. These objects can be PineAPPL grids (orange) or Evolution Kernel Operators (blue).

always a PineAPPL grid. Therefore, the same tools can be used on all of them.

The separation of the computation of the EKO and its convolution with the grid is convenient from a computational point of view. To illustrate the problem this separation solves, consider two possible scenarios:

- studies on the variation of $\alpha_s(M_Z)$ [29] which require only the recalculation of EKOs, but not the grids (Note that in Eq. (1) the strong coupling is factored out)

- studies on the variation of $M_W$ which require only the recalculation of grids, but not the EKOs.

The source code can be inspected from its repository

https://github.com/NNPDF/pineko

and the associated documentation inspected at

https://pineko.readthedocs.io.

## 4. Application: K-factors vs. exact predictions

As an application of the previously presented tools, we have integrated Vrap [24] into `pinefarm` and interfaced it to `PineAPPL` to produce FK tables for fixed-target Drell–Yan observables (FTDY) with up to next-to-next-to-leading order (NNLO) precision in the strong coupling. The step-by-step guide for the implementation of these results with the latest version of the `pineline` is documented at

https://nnpdf.github.io/pineline/examples/vrap

where only the last step is specific to the NNPDF framework.

In the following we use the framework presented in this paper and the steps outlined in the tutorial above to produce fits similar to NNPDF4.0 [2], which however differ in their treatment of predictions for the FTDY datasets: E605 [51], E866 [52, 53] and SeaQuest [54].

In particular, we change these predictions

1. to include *only NLO*,
2. to include NNLO *approximately* as K-factors (as in NNPDF4.0) and finally
3. to include NNLO *exactly* by using interpolation grids.

We note that the bulk of the hadron–hadron collider data (in particular all Drell–Yan Z and W production at the LHC) in all PDF fits are still limited to NNLO K-factors. K-factors are known to suffer from accidental cancellations between different partonic channels [55] and therefore they should be replaced by interpolation grids to produce a truly NNLO-accurate PDF fit. However, their use is widespread when studying complex observables for which the computation of exact NNLO prediction as a grid might be very difficult, computationally expensive, or simply not publicly available.

Fig. 2 shows the result of a fit including FTDY datasets only at NLO QCD (green), normalized to the results of a fit with exact NNLO QCD predictions (orange). In Fig. 3 we address the impact of including the NNLO contribution to the predictions in two different ways: exactly at NNLO (orange) and approximated by multiplying the NLO results by a bin-dependent K-factor (green).

In the particular case of FTDY we note already in Fig. 2 that the effect of NNLO corrections is constrained
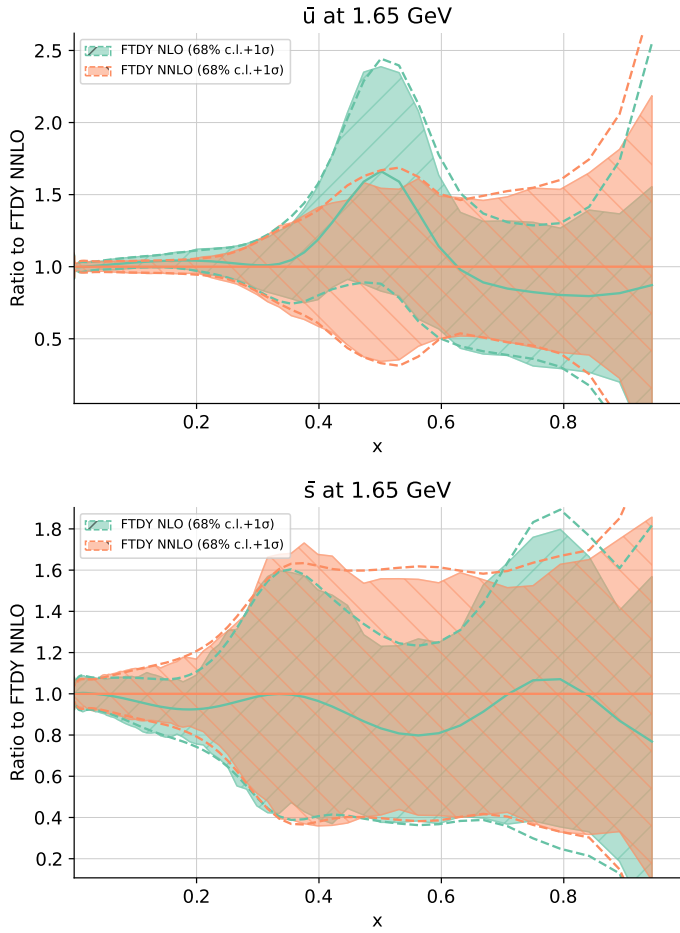
Figure 2: Comparison of PDF fits with and without NNLO contributions for FTDY in the determination. In both cases all other datasets are included at NNLO, the only difference between them is the exact NNLO contribution for FTDY.
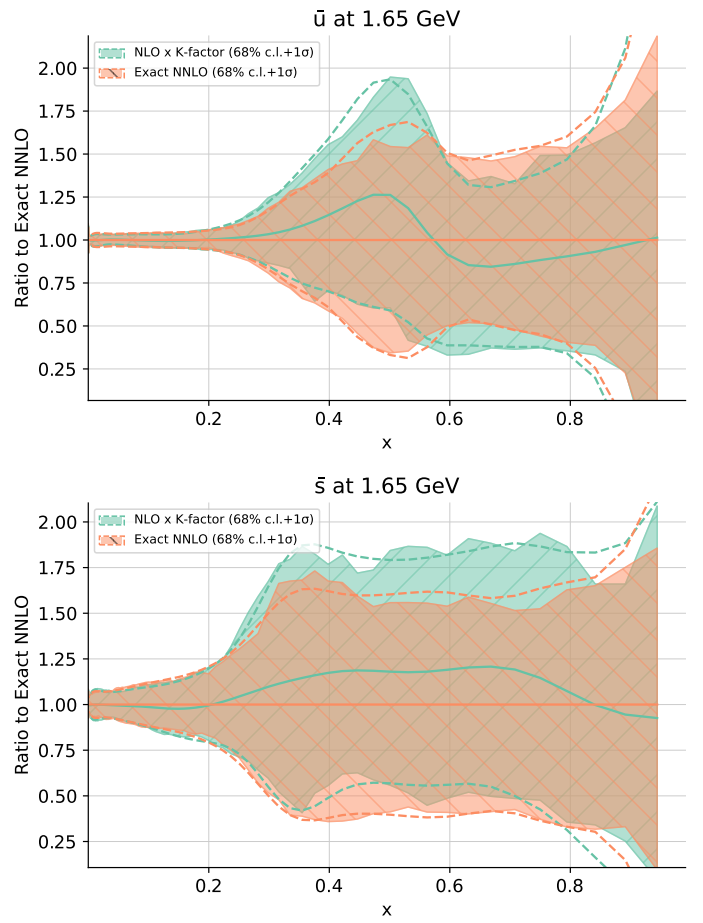
Figure 3: Comparison of PDF fits in which the FTDY datasets are included up to NNLO, including the exact predictions in the FK tables up to NNLO (orange) or up to NLO with K-factors (green). The orange fit corresponds to that of Fig. 2.
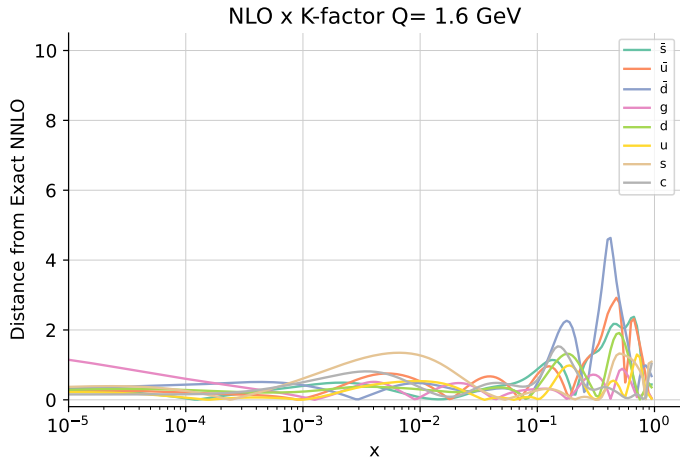
**NLO x K-factor Q= 1.6 GeV**

Figure 4: Distance plots between using the exact NNLO calculation and the K-factors as computed per Eq. (48) of Ref. [56]. A distance of 10 units corresponds in this case to a $1\sigma$ difference between the two PDF sets.

to a small portion of the PDF space. In Fig. 3 we can see that the effect of performing a fit with K-factors does move the fit in the direction expected from Fig. 2, but that the K-factors are not able to fully capture the nuances of the NNLO contribution. A similar behavior is shown by the plot of the $\bar{s}$ PDF in the same figures. These contributions are however compatible within uncertainties and the impact of using the K-factor approximation in this case is negligible. The quantitive difference between PDFs fitted from exact NNLO contributions or the K-factors is shown in Fig. 4; the difference is never significant and stay well below half a $\sigma$.

This is just one example of a phenomenological study facilitated by the framework presented in this paper. From a single run of Vrap we have been able to extract NLO, NLO×K-factor and NNLO (QCD) predictions. All of these predictions have been evolved to FK tables using the same NNLO EKOs, producing three different FK tables for three different fits. On the `pineline` site (`https://nnpdf.github.io/pineline`) the reader can find a tutorial for the reproduction of these results.

## 5. Conclusions and outlook

In this paper we described `pineline`, which is a collection of tools that includes `pinefarm` and `pineko`. The program `pinefarm` uses existing generators like `vrap`, `yadism` or `Madgraph5_aMC@NLO` to generate PineAPPL interpolation grids, which in turn can be converted to FK tables with `pineko`, which uses `EKO`. The produced objects are PineAPPL grids and store theory predictions independently from their PDFs, so that convolutions with arbitrary PDFs can be done near instantaneously after generation. The grids are useful for phenomenological studies, and we have shown an application in which we estimate the effect of replacing NNLO QCD K-factors by the exact calculation.

We want to highlight that the PineAPPL grids produced with `pinefarm` can be used by any external tool which includes an interface to said grids. One such example that has recently implemented this interface is the xFitter framework [57–60]. In Ref. [61], the authors present a phenomenological study making use of the software and tools presented in this paper for PDF fits with both the NNPDF and xFitter frameworks.

Below we name a few more applications for which we expect this framework to be beneficial due to the possibility of performing systematic studies of the effect of theory settings and theory predictions in PDF studies:

- we need to consistently account for theory uncertainties [40], coming either from the hard scattering process or the PDF evolution, and propagate these additional constraints into the final PDF delivery.

- Furthermore, it seems necessary to increase the perturbative order to next-to-next-to-next-to leading order (N3LO) [62, 63] to match the experimental precision.

- Finally, we need to consider the interplay of QCD and QED [64–66] and eventually consider EW corrections in a PDF determination.

In addition, the understanding of the impact of PDF uncertainties on beyond standard model searches [67] is fundamental in the hunt for new physics searches.

The framework is not restricted to the case of unpolarized proton PDF determination, but can already be applied to the extraction of other factorizable objects. Specifically, the extraction of transverse-momentum dependent PDFs [1, 68, 69] as well as the extraction of fragmentation functions [70] can be facilitated with the interpolation grids produced by this pipeline. With the advent of the EIC projects [27, 28] the refined determination of nuclear and polarized PDF [71] will also become available.

The framework also provides a standardized way to compare theory setting in different PDF groups, and allows an easy benchmark between the respective settings [72].

## Acknowledgments

## Appendix A. PineAPPL command-line interface (CLI) examples

In this section we give an overview of what can be done with the generated interpolation grids/FK tables from pine-

```
$ pineappl import applfast-h1-incjets-fnlo-arxiv-1611.03
    421-xsec000.tab.gz import.pineappl.lz4 CT18NNLO
b  PineAPPL fastNLO   rel. diff  svmaxreldiff
-+--------+--------+----------+------------
0  1.125e3  1.125e3  2.665e-15    7.883e-15
1  3.822e2  3.822e2  1.110e-15    8.216e-15
2  8.829e1  8.829e1  3.331e-15    5.773e-15
3  1.586e1  1.586e1 -3.997e-15    1.277e-14
4  2.725e0  2.725e0  4.663e-15    9.215e-15
5 4.848e-1 4.848e-1 -2.665e-15    1.310e-14
```

Figure A.5: Conversion of a flexible-scale fastNLO table into a PineAPPL grid. The output shows five columns: 1) the bin index from 0 to 5, 2) the result after the conversion, using the given PDF CT18NNLO, 3) the result using the PDF calculated from the fastNLO table, 4) the relative difference between the two previous results and finally 5) the maximum relative difference between between the two results which are scale varied. The relative differences come from differences in the fifteenth digit, as expected from double-precision results.

line with the PineAPPL's command-line interface (CLI) `pineappl`. See

https://nnpdf.github.io/pineappl/

for instructions on how to install it, and for more comprehensive documentation and tutorials. A dedicated page for the CLI can be found at

https://nnpdf.github.io/pineappl/docs/
cli-tutorial.html.

### Appendix A.1. Importing APPLgrids and fastNLO tables

The `import` subcommand of `pineappl` is able to convert existing APPLgrids and fastNLO tables into PineAPPL's format (`pineappl.lz4`). An example of the conversion of a flexible-scale fastNLO table, generated by the APPLfast project [21] for jet production at HERA [73] and downloaded from Ploughshare [37], is shown in Fig. A.5.

### Appendix A.2. Convoluting grids with PDF sets

Convolution of grids with a single or more PDF sets is done with `convolute`, as shown in the example in Fig. A.6. The interpolation grid corresponds to a measurement of the cross section of a single antilepton differentially in its pseudorapidity at LHCb [74].

### Appendix A.3. Other subcommands

PineAPPL's CLI `pineappl` offers many more subcommands that allow to convolute interpolation grids with PDFs in different ways: apart from `convolute` it can show the results separately by perturbative orders (`orders`), to read off the impact of higher-order corrections, and separately by channel (`channel`) to read off the hierarchy of which partonic initial states are most important. PDF and scale uncertainties can be calculated by `uncert`, and `pull` can be used to understand the differences of two PDFs: how large and significant they are and from which channels they come from. Two interpolation grids can be compared against each other using `diff`, `merge` combines two

```
$ wget https://github.com/NNPDF/pineapplgrids/raw/master
    /LHCB_WP_7TEV.pineappl.lz4
$ pineappl convolute LHCB_WP_7TEV.pineappl.lz4 NNPDF40_
    nnlo_as_01180 CT18NNLO
b   etal      dsig/detal        CT18NNLO
    []          [pb]             [pb]       [%]
-+----+----+-----------+-----------+-----
0    2 2.25 7.8847492e2 7.7526895e2 -1.67
1 2.25  2.5 7.2061556e2 7.1092145e2 -1.35
2  2.5 2.75 6.2526363e2 6.1876958e2 -1.04
3 2.75    3 5.0385884e2 5.0017809e2 -0.73
4    3 3.25 3.7400170e2 3.7228440e2 -0.46
5 3.25  3.5 2.5300890e2 2.5236943e2 -0.25
6  3.5    4 1.1909464e2 1.1857770e2 -0.43
7    4  4.5 2.9004607e1 2.7740964e1 -4.36
```

Figure A.6: Convolution of a PineAPPL grid with two PDF sets. This shows the following columns: 1) the bin index from 0 to 7, 2) the left and 3) the right limit of each bin and 4) the prediction for each bin using the first PDF set. For each additional PDF set given the 5) absolute numbers are shown and 6) the relative differences to the first PDF set. Using metadata the first (and only) bin dimension is denoted as 'etal', the pseudorapidity of the antilepton, which does not have a unit, shown by the empty square-brackets. The prediction is shown as 'dsig/detal', so the cross section differentially in the pseudorapidity of the antilepton, in units of picobarn.

or more grids by combining them correctly and `plot` writes a Python plot script, which graphically display most of the information given numerically above. Finally, the subcommands `read` and `write` can be used to read out data and metadata from a grid and perform operations and set metadata, respectively.

## References

[1] S. Amoroso, et al., Snowmass 2021 whitepaper: Proton structure at the precision frontier (3 2022). arXiv:2203.13923.

[2] R. D. Ball, et al., The path to proton structure at 1% accuracy, Eur. Phys. J. C 82 (5) (2022) 428. arXiv:2109.02653, doi:10.1140/epjc/s10052-022-10328-7.

[3] S. Bailey, T. Cridge, L. A. Harland-Lang, A. D. Martin, R. S. Thorne, Parton distributions from LHC, HERA, Tevatron and fixed target data: MSHT20 PDFs, Eur. Phys. J. C 81 (4) (2021) 341. arXiv:2012.04684, doi:10.1140/epjc/s10052-021-09057-0.

[4] T.-J. Hou, et al., New CTEQ global analysis of quantum chromodynamics with high-precision data from the LHC, Phys. Rev. D 103 (1) (2021) 014013. arXiv:1912.10053, doi:10.1103/PhysRevD.103.014013.

[5] S. Alekhin, J. Blümlein, S. Moch, R. Placakyte, Parton distribution functions, $\alpha_s$, and heavy-quark masses for LHC Run II, Phys. Rev. D 96 (1) (2017) 014011. arXiv:1701.05838, doi:10.1103/PhysRevD.96.014011.

[6] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons, The fair guiding principles for scientific data management and stewardship, Scientific Data 3 (1) (2016) 160018. doi:10.1038/sdata.2016.18.
URL https://doi.org/10.1038/sdata.2016.18

[7] V. Bertone, S. Carrazza, J. Rojo, APFEL: A PDF Evolution Library with QED corrections, Comput. Phys. Commun. 185 (2014) 1647–1668. `arXiv:1310.1394`, `doi:10.1016/j.cpc.2014.03.007`.

[8] S. Catani, M. Grazzini, An NNLO subtraction formalism in hadron collisions and its application to Higgs boson production at the LHC, Phys. Rev. Lett. 98 (2007) 222002. `arXiv:hep-ph/0703012`, `doi:10.1103/PhysRevLett.98.222002`.

[9] S. Catani, L. Cieri, G. Ferrera, D. de Florian, M. Grazzini, Vector boson production at hadron colliders: a fully exclusive QCD calculation at NNLO, Phys. Rev. Lett. 103 (2009) 082001. `arXiv:0903.2120`, `doi:10.1103/PhysRevLett.103.082001`.

[10] R. Gavin, Y. Li, F. Petriello, S. Quackenbush, FEWZ 2.0: A code for hadronic Z production at next-to-next-to-leading order, Comput. Phys. Commun. 182 (2011) 2388–2403. `arXiv:1011.3540`, `doi:10.1016/j.cpc.2011.06.008`.

[11] R. Gavin, Y. Li, F. Petriello, S. Quackenbush, W Physics at the LHC with FEWZ 2.1, Comput. Phys. Commun. 184 (2013) 208–214. `arXiv:1201.5896`, `doi:10.1016/j.cpc.2012.09.005`.

[12] Y. Li, F. Petriello, Combining QCD and electroweak corrections to dilepton production in FEWZ, Phys. Rev. D 86 (2012) 094034. `arXiv:1208.5967`, `doi:10.1103/PhysRevD.86.094034`.

[13] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, M. Zaro, The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations, JHEP 07 (2014) 079. `arXiv:1405.0301`, `doi:10.1007/JHEP07(2014)079`.

[14] R. Frederix, S. Frixione, V. Hirschi, D. Pagani, H. S. Shao, M. Zaro, The automation of next-to-leading order electroweak calculations, JHEP 07 (2018) 185. `arXiv:1804.10017`, `doi:10.1007/JHEP07(2018)185`.

[15] J. M. Campbell, R. K. Ellis, An Update on vector boson pair production at hadron colliders, Phys. Rev. D 60 (1999) 113006. `arXiv:hep-ph/9905386`, `doi:10.1103/PhysRevD.60.113006`.

[16] J. M. Campbell, R. K. Ellis, C. Williams, Vector boson pair production at the LHC, JHEP 07 (2011) 018. `arXiv:1105.0020`, `doi:10.1007/JHEP07(2011)018`.

[17] J. M. Campbell, R. K. Ellis, W. T. Giele, A Multi-Threaded Version of MCFM, Eur. Phys. J. C 75 (6) (2015) 246. `arXiv:1503.06182`, `doi:10.1140/epjc/s10052-015-3461-2`.

[18] J. Campbell, T. Neumann, Precision Phenomenology with MCFM, JHEP 12 (2019) 034. `arXiv:1909.09117`, `doi:10.1007/JHEP12(2019)034`.

[19] R. Boughezal, C. Focke, X. Liu, F. Petriello, W-boson production in association with a jet at next-to-next-to-leading order in perturbative QCD, Phys. Rev. Lett. 115 (6) (2015) 062002. `arXiv:1504.02131`, `doi:10.1103/PhysRevLett.115.062002`.

[20] A. Gehrmann-De Ridder, T. Gehrmann, E. W. N. Glover, A. Huss, T. A. Morgan, Precise QCD predictions for the production of a Z boson in association with a hadronic jet, Phys. Rev. Lett. 117 (2) (2016) 022001. `arXiv:1507.02850`, `doi:10.1103/PhysRevLett.117.022001`.

[21] D. Britzger, et al., Calculations for deep inelastic scattering using fast interpolation grid techniques at NNLO in QCD and the extraction of $\alpha_s$ from HERA data, Eur. Phys. J. C 79 (10) (2019) 845, [Erratum: Eur.Phys.J.C 81, 957 (2021)]. `arXiv:1906.05303`, `doi:10.1140/epjc/s10052-021-09688-3`.

[22] Z. Nagy, Three jet cross-sections in hadron hadron collisions at next-to-leading order, Phys. Rev. Lett. 88 (2002) 122003. `arXiv:hep-ph/0110315`, `doi:10.1103/PhysRevLett.88.122003`.

[23] M. Czakon, A. Mitov, Top++: A Program for the Calculation of the Top-Pair Cross-Section at Hadron Colliders, Comput. Phys. Commun. 185 (2014) 2930. `arXiv:1112.5675`, `doi:10.1016/j.cpc.2014.06.021`.

[24] C. Anastasiou, L. J. Dixon, K. Melnikov, F. Petriello, High precision QCD at hadron colliders: Electroweak gauge boson rapidity distributions at NNLO, Phys. Rev. D 69 (2004) 094008. `arXiv:hep-ph/0312266`, `doi:10.1103/PhysRevD.69.094008`.

[25] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, J. Winter, Event generation with SHERPA 1.1, JHEP 02 (2009) 007. `arXiv:0811.4622`, `doi:10.1088/1126-6708/2009/02/007`.

[26] J. Gao, L. Harland-Lang, J. Rojo, The Structure of the Proton in the LHC Precision Era, Phys. Rept. 742 (2018) 1–121. `arXiv:1709.04922`, `doi:10.1016/j.physrep.2018.03.002`.

[27] A. Accardi, et al., Electron Ion Collider: The Next QCD Frontier: Understanding the glue that binds us all, Eur. Phys. J. A 52 (9) (2016) 268. `arXiv:1212.1701`, `doi:10.1140/epja/i2016-16268-9`.

[28] D. P. Anderle, et al., Electron-ion collider in China, Front. Phys. (Beijing) 16 (6) (2021) 64701. `arXiv:2102.09222`, `doi:10.1007/s11467-021-1062-0`.

[29] S. Forte, Z. Kassabov, Why $\alpha_s$ cannot be determined from hadronic processes without simultaneously determining the parton distributions, Eur. Phys. J. C 80 (3) (2020) 182. `arXiv:2001.04986`, `doi:10.1140/epjc/s10052-020-7748-6`.

[30] T. Carli, D. Clements, A. Cooper-Sarkar, C. Gwenlan, G. P. Salam, F. Siegert, P. Starovoitov, M. Sutton, A posteriori inclusion of parton density functions in NLO QCD final-state calculations at hadron colliders: The APPLGRID Project, Eur. Phys. J. C 66 (2010) 503–524. `arXiv:0911.2985`, `doi:10.1140/epjc/s10052-010-1255-0`.

[31] D. Britzger, K. Rabbertz, F. Stober, M. Wobisch, New features in version 2 of the fastNLO project, in: 20th International Workshop on Deep-Inelastic Scattering and Related Subjects, 2012, pp. 217–221. `arXiv:1208.3641`, `doi:10.3204/DESY-PROC-2012-02/165`.

[32] C. Schwan, A. Candido, F. Hekhorn, S. Carrazza, Nnpdf/pineappl: v0.5.9 (Jan. 2023). `doi:10.5281/zenodo.7499507`.
URL https://doi.org/10.5281/zenodo.7499507

[33] L. Del Debbio, N. P. Hartland, S. Schumann, MCgrid: projecting cross section calculations on grids, Comput. Phys. Commun. 185 (2014) 2115–2126. `arXiv:1312.4460`, `doi:10.1016/j.cpc.2014.03.023`.

[34] V. Bertone, R. Frederix, S. Frixione, J. Rojo, M. Sutton, aMC-fast: automation of fast NLO computations for PDF fits, JHEP 08 (2014) 166. `arXiv:1406.7693`, `doi:10.1007/JHEP08(2014)166`.

[35] S. Carrazza, E. R. Nocera, C. Schwan, M. Zaro, PineAPPL: combining EW and QCD corrections for fast evaluation of LHC processes, JHEP 12 (2020) 108. `arXiv:2008.12789`, `doi:10.1007/JHEP12(2020)108`.

[36] E. Maguire, L. Heinrich, G. Watt, HEPData: a repository for high energy physics data, J. Phys. Conf. Ser. 898 (10) (2017) 102006. `arXiv:1704.05473`, `doi:10.1088/1742-6596/898/10/102006`.

[37] Ploughshare.
URL http://ploughshare.web.cern.ch/

[38] R. D. Ball, et al., An open-source machine learning framework for global analyses of parton distributions, Eur. Phys. J. C 81 (10) (2021) 958. `arXiv:2109.02671`, `doi:10.1140/epjc/s10052-021-09747-9`.

[39] R. D. Ball, et al., Parton distributions for the LHC Run II, JHEP 04 (2015) 040. `arXiv:1410.8849`, `doi:10.1007/JHEP04(2015)040`.

[40] R. Abdul Khalek, et al., Parton Distributions with Theory Uncertainties: General Formalism and First Phenomenological Studies, Eur. Phys. J. C 79 (11) (2019) 931. `arXiv:1906.10698`, `doi:10.1140/epjc/s10052-019-7401-4`.

[41] A. Candido, F. Hekhorn, G. Magni, N3pdf/yadism: Fonll-b (Feb. 2022). `doi:10.5281/zenodo.6285149`.
URL https://doi.org/10.5281/zenodo.6285149

[42] M. Grazzini, S. Kallweit, M. Wiesemann, Fully differential NNLO computations with MATRIX, Eur. Phys. J. C 78 (7) (2018) 537. `arXiv:1711.06631`, `doi:10.1140/epjc/s10052-018-5771-7`.

[43] G. Altarelli, G. Parisi, Asymptotic Freedom in Parton Language, Nucl. Phys. B126 (1977) 298–318. `doi:10.1016/0550-3213(77)90384-4`.

[44] V. N. Gribov, L. N. Lipatov, Deep inelastic e p scattering in

perturbation theory, Sov. J. Nucl. Phys. 15 (1972) 438–450, [Yad. Fiz.15,781(1972)].

[45] Y. L. Dokshitzer, Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics., Sov. Phys. JETP 46 (1977) 641–653, [Zh. Eksp. Teor. Fiz.73,1216(1977)].

[46] A. Candido, F. Hekhorn, G. Magni, N3pdf/eko: Paper (Mar. 2022). doi:10.5281/zenodo.6340153. URL https://doi.org/10.5281/zenodo.6340153

[47] A. Candido, F. Hekhorn, G. Magni, EKO: evolution kernel operators, Eur. Phys. J. C 82 (10) (2022) 976. arXiv:2202.02338, doi:10.1140/epjc/s10052-022-10878-w.

[48] A. Vogt, Efficient evolution of unpolarized and polarized parton distributions with QCD-PEGASUS, Comput. Phys. Commun. 170 (2005) 65–92. arXiv:hep-ph/0408244, doi:10.1016/j.cpc.2005.03.103.

[49] M. Botje, QCDNUM: Fast QCD Evolution and Convolution, Comput.Phys.Commun. 182 (2011) 490–532. arXiv:1005.1481, doi:10.1016/j.cpc.2010.10.020.

[50] V. Bertone, APFEL++: A new PDF evolution library in C++, PoS DIS2017 (2018) 201. arXiv:1708.00911, doi:10.22323/1.297.0201.

[51] G. Moreno, et al., Dimuon production in proton - copper collisions at $\sqrt{s}$ = 38.8-GeV, Phys. Rev. D43 (1991) 2815–2836. doi:10.1103/PhysRevD.43.2815.

[52] J. C. Webb, et al., Absolute Drell-Yan dimuon cross sections in 800-GeV/c p p and p d collisions (2003). arXiv:hep-ex/0302019.

[53] R. S. Towell, et al., Improved measurement of the anti-d/anti-u asymmetry in the nucleon sea, Phys. Rev. D64 (2001) 052002. arXiv:hep-ex/0103030, doi:10.1103/PhysRevD.64.052002.

[54] J. Dove, et al., The asymmetry of antimatter in the proton, Nature 590 (7847) (2021) 561–565. arXiv:2103.04024, doi:10.1038/s41586-021-03282-z.

[55] C. Duhr, F. Dulat, B. Mistlberger, Drell-Yan Cross Section to Third Order in the Strong Coupling Constant, Phys. Rev. Lett. 125 (17) (2020) 172001. arXiv:2001.07717, doi:10.1103/PhysRevLett.125.172001.

[56] R. D. Ball, et al., Parton distributions for the LHC Run II, JHEP 04 (2015) 040. arXiv:1410.8849, doi:10.1007/JHEP04(2015)040.

[57] S. Alekhin, et al., HERAFitter, Eur. Phys. J. C 75 (7) (2015) 304. arXiv:1410.4412, doi:10.1140/epjc/s10052-015-3480-z.

[58] V. Bertone, M. Botje, D. Britzger, et al., xFitter 2.0.0: An open source QCD fit framework, PoS DIS2017 (2018) 203. arXiv:1709.01151, doi:10.22323/1.297.0203.

[59] H. Abdolmaleki, et al., xFitter: An Open Source QCD Analysis Framework. A resource and reference document for the Snowmass study, 2022. arXiv:2206.12465.

[60] xFitter Developers' Team. [link]. URL https://www.xfitter.org/xFitter/

[61] J. M. Cruz-Martinez, M. Fieg, T. Giani, P. Krack, T. Mäkelä, T. Rabemananjara, J. Rojo, The LHC as a Neutrino-Ion Collider (9 2023). arXiv:2309.09581.

[62] F. Caola, W. Chen, C. Duhr, X. Liu, B. Mistlberger, F. Petriello, G. Vita, S. Weinzierl, The Path forward to N$^3$LO, in: 2022 Snowmass Summer Study, 2022. arXiv:2203.06730.

[63] J. McGowan, T. Cridge, L. A. Harland-Lang, R. S. Thorne, Approximate N$^3$LO Parton Distribution Functions with Theoretical Uncertainties: MSHT20aN$^3$LO PDFs (7 2022). arXiv:2207.04739.

[64] K. Xie, T. J. Hobbs, T.-J. Hou, C. Schmidt, M. Yan, C. P. Yuan, Photon PDF within the CT18 global analysis, Phys. Rev. D 105 (5) (2022) 054006. arXiv:2106.10299, doi:10.1103/PhysRevD.105.054006.

[65] T. Cridge, L. A. Harland-Lang, A. D. Martin, R. S. Thorne, QED parton distribution functions in the MSHT20 fit, Eur. Phys. J. C 82 (1) (2022) 90. arXiv:2111.05357, doi:10.1140/epjc/s10052-022-10028-2.

[66] V. Bertone, S. Carrazza, N. P. Hartland, J. Rojo, Illuminating the photon content of the proton within a global PDF analysis, SciPost Phys. 5 (1) (2018) 008. arXiv:1712.07053, doi:10.21468/SciPostPhys.5.1.008.

[67] R. D. Ball, A. Candido, S. Forte, F. Hekhorn, E. R. Nocera, J. Rojo, C. Schwan, Parton distributions and new physics searches: the Drell–Yan forward–backward asymmetry as a case study, Eur. Phys. J. C 82 (12) (2022) 1160. arXiv:2209.08115, doi:10.1140/epjc/s10052-022-11133-y.

[68] A. Bacchetta, V. Bertone, C. Bissolotti, G. Bozzi, M. Cerutti, F. Piacenza, M. Radici, A. Signori, Unpolarized transverse momentum distributions from a global fit of Drell-Yan and semi-inclusive deep-inelastic scattering data, JHEP 10 (2022) 127. arXiv:2206.07598, doi:10.1007/JHEP10(2022)127.

[69] M. Cerutti, L. Rossi, S. Venturini, A. Bacchetta, V. Bertone, C. Bissolotti, M. Radici, Extraction of pion transverse momentum distributions from Drell-Yan data, Phys. Rev. D 107 (1) (2023) 014014. arXiv:2210.01733, doi:10.1103/PhysRevD.107.014014.

[70] R. Abdul Khalek, V. Bertone, A. Khoudli, E. R. Nocera, Pion and kaon fragmentation functions at next-to-next-to-leading order, Phys. Lett. B 834 (2022) 137456. arXiv:2204.10331, doi:10.1016/j.physletb.2022.137456.

[71] J. J. Ethier, E. R. Nocera, Parton Distributions in Nucleons and Nuclei, Ann. Rev. Nucl. Part. Sci. 70 (2020) 43–76. arXiv:2001.07722, doi:10.1146/annurev-nucl-011720-042725.

[72] R. D. Ball, et al., The PDF4LHC21 combination of global PDF fits for the LHC Run III, J. Phys. G 49 (8) (2022) 080501. arXiv:2203.05506, doi:10.1088/1361-6471/ac7216.

[73] V. Andreev, et al., Measurement of Jet Production Cross Sections in Deep-inelastic ep Scattering at HERA, Eur. Phys. J. C 77 (4) (2017) 215, [Erratum: Eur.Phys.J.C 81, 739 (2021)]. arXiv:1611.03421, doi:10.1140/epjc/s10052-017-4717-9.

[74] R. Aaij, et al., Measurement of the forward Z boson production cross-section in pp collisions at $\sqrt{s}$ = 7 TeV, JHEP 08 (2015) 039. arXiv:1505.07024, doi:10.1007/JHEP08(2015)039.