Eur. Phys. J. C (2021) 81:958
https://doi.org/10.1140/epjc/s10052-021-09747-9

THE EUROPEAN PHYSICAL JOURNAL C

Special Article - Tools for Experiment and Theory

An open-source machine learning framework for global analyses of parton distributions

NNPDF Collaboration

Richard D. Ball[1], Stefano Carrazza[2], Juan Cruz-Martinez[2], Luigi Del Debbio[1], Stefano Forte[2], Tommaso Giani[7,8], Shayan Iranipour[3], Zahari Kassabov[3], Jose I. Latorre[4,5,6], Emanuele R. Nocera[1,8], Rosalyn L. Pearson[1], Juan Rojo[7,8], Roy Stegeman[2], Christopher Schwan[2], Maria Ubiali[3,a], Cameron Voisey[9], Michael Wilson[1]

[1] The Higgs Centre for Theoretical Physics, University of Edinburgh, JCMB, KB, Mayfield Rd, Edinburgh EH9 3JZ, Scotland, UK
[2] Tif Lab, Dipartimento di Fisica, Università di Milano and INFN, Sezione di Milano, Via Celoria 16, 20133 Milan, Italy
[3] DAMTP, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, UK
[4] Quantum Research Centre, Technology Innovation Institute, Abu Dhabi, United Arab Emirates
[5] Center for Quantum Technologies, National University of Singapore, Singapore, Singapore
[6] Qilimanjaro Quantum Tech, Barcelona, Spain
[7] Department of Physics and Astronomy, VU Amsterdam, 1081 HV Amsterdam, The Netherlands
[8] Nikhef Theory Group, Science Park 105, 1098 XG Amsterdam, The Netherlands
[9] Cavendish Laboratory, University of Cambridge, Cambridge CB3 0HE, UK

Received: 17 September 2021 / Accepted: 10 October 2021 / Published online: 30 October 2021
© The Author(s) 2021

Abstract We present the software framework underlying the NNPDF4.0 global determination of parton distribution functions (PDFs). The code is released under an open source licence and is accompanied by extensive documentation and examples. The code base is composed by a PDF fitting package, tools to handle experimental data and to efficiently compare it to theoretical predictions, and a versatile analysis framework. In addition to ensuring the reproducibility of the NNPDF4.0 (and subsequent) determination, the public release of the NNPDF fitting framework enables a number of phenomenological applications and the production of PDF fits under user-defined data and theory assumptions.

Contents

[a] e-mail: M.Ubiali@damtp.cam.ac.uk (corresponding author)

1 Introduction

The success of the ambitious programme of the upcoming Run III at the LHC and its subsequent High-Luminosity upgrade [1,2] relies on achieving the highest possible accuracy not only in the experimental measurements but also in the corresponding theoretical predictions. A key component of the latter are the parton distribution functions (PDFs), which parametrize the quark and gluon substructure of the colliding protons [3,4]. PDFs are dictated by non-perturbative QCD dynamics and hence must be phenomenologically extracted by matching a wide range of experimental data with the corresponding theoretical predictions. The determination of PDFs and their uncertainties requires a robust statistical framework which minimises unnecessary assumptions while implementing known theoretical constraints such as QCD evolution, sum rules, positivity, and integrability.

Recently, a new family of global PDF analyses has been presented by the NNPDF Collaboration: NNPDF4.0 [5]. This updated PDF determination framework supersedes its predecessor NNPDF3.1 [6] by improving on all relevant aspects, from the experimental input and theoretical constraints to the optimisation methodology and the validation of results. As with previous NNPDF releases, the NNPDF4.0 PDFs are made publicly available via the standard LHAPDF interface [7]. However, until now only the outcome of the NNPDF fits

(the LHAPDF interpolation grid files) was released, while the code itself remained private. This situation implied that the only option to produce tailored variants of the NNPDF analyses was by requesting them to the developers, and further that results were not reproducible by external parties. Another limitation of private PDF codes is that benchmarking studies, such as those carried out by the PDF4LHC working group [8,9], become more convoluted due to the challenge in disentangling the various components that determine the final outcome.

Motivated by this state of affairs, as well as by the principles of Open and FAIR [10] (findable, accessible, interoperable and reusable) Science, in this work we describe the public release of the complete software framework [11] underlying the NNPDF4.0 global determination together with user-friendly examples and an extensive documentation. In addition to the fitting code itself, this release includes the original and filtered experimental data, the fast NLO interpolation grids relevant for the computation of hadronic observables, and whenever available the bin-by-bin next-to-next-to-leading order (NNLO) QCD and next-to-leading (NLO) electroweak K-factors for all processes entering the fit. Furthermore, the code comes accompanied by a battery of plotting, statistical, and diagnosis tools providing the user with an extensive characterisation of the PDF fit output.

The availability of the NNPDF open-source code, along with its detailed online documentation, will enable users to perform new PDF analyses based on the NNPDF methodology and modifications thereof. Some examples of potential applications include assessing the impact of new measurements in the global fit; producing variants based on reduced datasets, carrying out PDF determinations with different theory settings, e.g. as required for studies of α_s or heavy quark mass sensitivity, or with different electroweak parameters; estimating the impact on the PDFs of theoretical constraints and calculations e.g. from non-perturbative QCD models [12] or lattice calculations [13,14]; and quantifying the role of theoretical uncertainties from missing higher orders to nuclear effects. One could also deploy the NNPDF code as a toolbox to pin down the possible effects of beyond the Standard Model physics at the LHC, such as Effective Field Theory corrections in high-p_T tails [15,16] or modified DGLAP evolution from new BSM light degrees of freedom [17]. Furthermore, while the current version of the NNPDF code focuses on unpolarised parton distributions, its modular and flexible infrastructure makes it amenable to the determination of closely related pace* non-perturbative collinear QCD quantities such as polarised PDFs, nuclear PDFs, fragmentation functions, or even the parton distributions of mesons like pions and kaons [18].

It should be noted that some of the functionalities described above are already available within the open source QCD fit framework `xFitter` [19,20]. The NNPDF code offers complementary functionalities as compared to those in `xFitter`, in particular by means of state-of-the-art machine learning tools for the PDF parametrisation, robust methods for uncertainty estimate and propagation, a wider experimental dataset, an extensive suite of statistical validation and plotting tools, the possibility to account for generic theoretical uncertainties, and an excellent computational performance which makes possible full-fledged global PDF fits in less than one hour.

The main goal of this paper is to summarise the key features of the NNPDF code and to point the interested reader to the online documentation, in which the code is presented in detail and which, importantly, is kept up-to-date as the code continues to be developed and improved. First, in Sect. 2 we describe the structure of the code and its main functionalities, including the relevant options. The framework used to analyse the outcome of a PDF fit is described in Sect. 3, while in Sect. 4 we describe a few examples of possible applications for which users may wish to use the code. We conclude and summarise some possible directions of future development in Sect. 5.

2 Code structure

The open-source `NNPDF` framework enables performing global QCD analyses of lepton-proton(nucleus) and proton-(anti)proton scattering data in terms of the NNPDF4.0 methodology described in [5]. The code is publicly available from its GitHub repository

https://github.com/NNPDF/

and is accompanied by an extensive, continuously updated, online documentation

https://docs.nnpdf.science/

In this section, we describe the structure of the code and we present a high-level description of its functionalities. We invite the reader to consult the documentation for details on its usage.

The workflow for the NNPDF code is illustrated in Fig. 1. The NNPDF code is composed of the following main packages:

The `buildmaster` experimental data formatter
A `C++` code which transforms the original measurements provided by the experimental collaborations, e.g. via HepData [21], into a standard format that is tailored for PDF fitting.

In particular, the code allows for a flexible handling of experimental systematic uncertainties allowing for different treatments of the correlated systematic uncertainties [22,23].

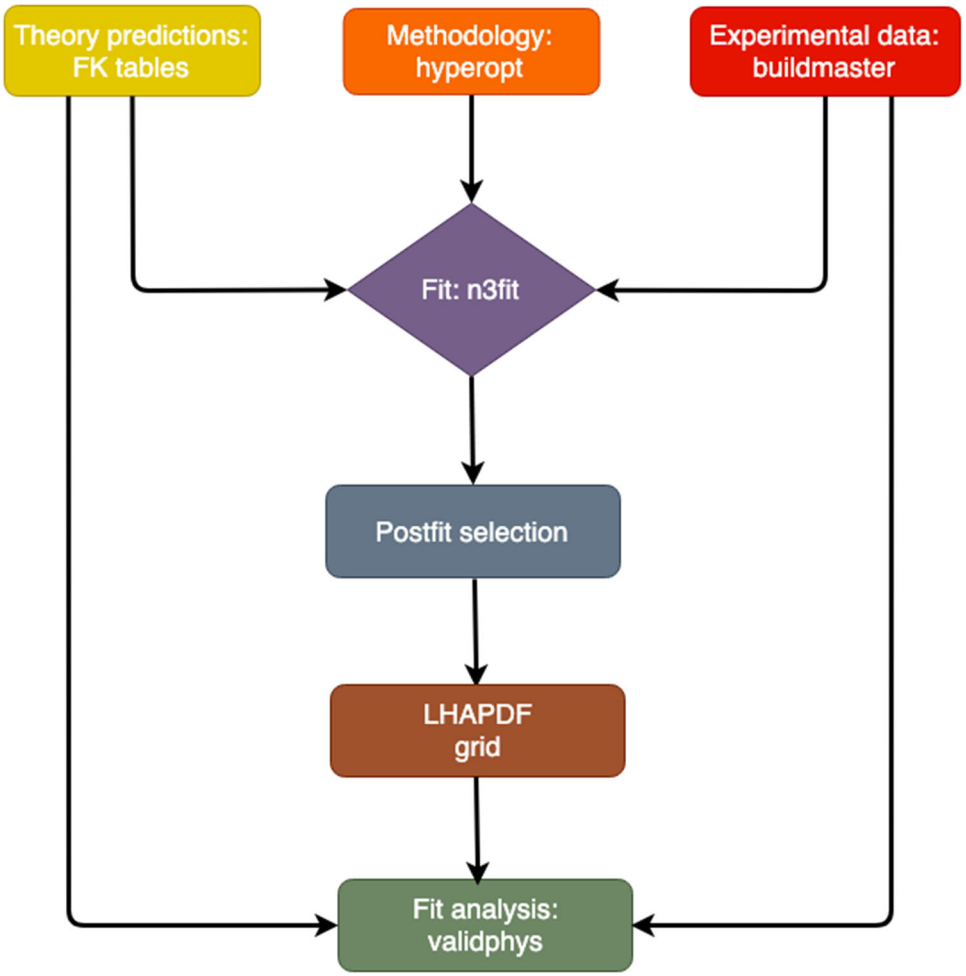

Fig. 1 Schematic of the NNPDF code. The three main inputs are the theoretical calculations, encoded in terms of the precomputed FK-tables, the methodological settings as determined by the hyperopt procedure, and the experimental data in the common buildmaster format. The PDFs are fitted using n3fit, and following a postfit selection the outcome is stored in the LHAPDF grid format. Finally, a thorough characterisation of the results is carried out by the validphys framework

The APFELcomb interpolation table generator
This code takes hard-scattering partonic matrix element interpolators from APPLgrid [24] and FastNLO [25] (for hadronic processes) and APFEL [26] (for DIS structure functions) and combines them with the QCD evolution kernels provided by APFEL to construct the fast interpolation grids called FK-tables [27]. In this way, physical observables can be evaluated in a highly efficient manner as a tensor sum of FK-tables with a grid of PDFs at an initial parametrisation scale Q_0. APFELcomb also handles NNLO QCD and/or NLO electroweak K-factors when needed.

Theory predictions can be generated configuring a variety of options, such as the perturbative order (currently up to NNLO), the values of the heavy quark masses, the electroweak parameters, the maximum number of active flavours, and the variable-flavour-number scheme used to account for the effects of the heavy quark masses in the DIS structure functions. The FK-tables resulting from each choice are associated to a database entry trough a theory id, which allows to quickly identify them them.

The n3fit fitting code
This code implements the fitting methodology described in [5,28] as implemented in the TensorFlow framework [29]. The n3fit library allows for a flexible specification of the neural network model adopted to parametrise the PDFs, whose settings can be selected automatically via the built-in hyperoptimisation tooling [30]. These include the neural network type and architecture, the activation functions, and the initialisation strategy; the choice of optimiser and of its corresponding parameters; and hyperparameters related to the implementation in the fit of theoretical constraints such as PDF positivity [31] and integrability. The settings for a PDF fit are input via a declarative run card. Using these settings, n3fit finds the values of the neural network parameters, corresponding to the PDF at initial scale which describe the input data. Following a post-fit selection and PDF evolution step, the final output consists of an LHAPDF grid corresponding to the best fit PDF as well as metadata on the fit performance.

The libnnpdf C++ legacy code
A C++ library which contains common data structures

together with the fitting code used to produce the NNPDF3.0 and NNPDF3.1 analyses [6,32–35].

The availability of the `libnnpdf` guarantees strict backwards compatibility of the NNPDF framework and the ability to benchmark the current methodology against the previous one.

To facilitate the interaction between the NNPDF `C++` and `Python` codebases, we have developed `Python` wrappers using the `SWIG` [36] library.

The `validphys` analysis framework

A package allowing to analyse and plot data related to the NNPDF fit structures and I/O capabilities to other elements of the code base. The `validphys` framework is discussed in detail in Sect. 3.

Complementing these main components, the NNPDF framework also contains a number of additional, ever-evolving, tools which are described in the online documentation.

Development workflow. The NNPDF code adopts a development workflow compliant with best practices in professionally developed software projects. Specifically, every code modification undergoes code review and is subjected to a suite of automated continuous integration testing. Moreover, before merging into the main release branch, all relevant documentation is added alongside any new tests that may be relevant to the incoming feature. This feature ensures that a broad code coverage within the test suite is maintained.

Installation. The various software packages that compose the NNPDF fitting code can be installed via the binary packages provided by the `conda` interface, as described in

https://docs.nnpdf.science/get-started/installation.html

The binary distribution allows users to easily install the entire code suite alongside all relevant dependencies within an isolated environment, which is also compatible with the one that has been tested automatically. Consequently, PDF fits can be produced with a known fixed version of the code and all its dependencies, regardless of the machine where it is running, hence ensuring the reproducibility of the result. For the purposes of code development, it is also possible to set up an environment where the dependencies are the same but the code can be edited, allowing users to contribute to the open-source framework.

Input configuration. The settings that define the outcome of a NNPDF fit are specified by means of a run card written in `YAML`, a common human-readable data-serialisation language. The main elements of fit run cards are:

Input dataset: for each dataset, the user has to specify the NNPDF-internal string associated to it, the fraction of the data that goes into the training and validation subsets, and the inclusion of K-factors in the corresponding theoretical predictions.
The latter are assigned different naming conventions depending on their nature: NNLO QCD, NLO electroweak, heavy-quark mass corrections for neutrino DIS [37], or overall normalisation rescaling.
Correlations between common systematic uncertainties between different datasets are automatically taken into account.

Kinematical cuts: a declarative format that specifies the cuts applied to the experimental data, based on the kinematics of each data point and depending on the corresponding theory settings. The cuts can be based on simple relations between the kinematics of each data point, such as the usual $Q^2_{\rm min}$ and $W^2_{\rm min}$ cuts applied to DIS structure functions, some derived quantity such as the value of the lepton charge asymmetry in W decay data, or on more complex conditions such as retaining only points where the relative difference between NLO and NNLO predictions is below some threshold.
These kinematical cut configuration can either be specified directly in the run card or the built-in defaults can be used, and can be required for individual datasets or for types of processes instead.

Theory settings: the settings for theory predictions to be used in the fit, such as the perturbative order and the values of the coupling constants and of the quark masses, are specified an entry in the theory database, which in turn selects the set of `FK`-tables, to be used during the fit.
A wide range of `FK`-tables for the most commonly used theory settings are already available and can be installed using the NNPDF code, while tables corresponding to different settings can also be assembled by the user whenever required. The settings for the available entries of the theory database are specified in the online documentation.

Fitting strategy and hyperparameters: the user can specify via the run card a number of methodological settings that affect the optimisation, such as the minimisation algorithm with the corresponding parameters, the maximum training length, the neural network architecture and activation functions, and the choice of PDF fitting basis (e.g. using the evolution or the flavour basis).

These methodological settings can either be set by hand or taken from the result of a previous `hyperopt` run. Furthermore, random seeds can be configured to achieve different levels of correlation between Monte Carlo replicas across

fits, as required e.g. for the correlated replica method used in the $\alpha_s(m_Z)$ extraction of [38].

The user can additionally decide whether to save the weights of the neural networks during the fit or not, and whether to fit the Monte Carlo replicas or instead the central values of the experimental data.

Another choice accessible via the run card is whether to use real data or instead fit to pseudo-data generated from a known underlying PDFs, as required during a closure test [32,39].

PDF positivity and integrability: as described in [5], in the NNPDF4.0 determination one imposes theoretical requirements on the positivity and integrability of the fitted PDFs by means of the Lagrange multiplier method.

The user can then decide via the run card whether or not (or only partially) to impose these constraints on the PDFs, and if so define the initial values of the Lagrange multiplier weights.

Note that some of the parameters governing the implementation of these theory requirements can also be adjusted by means of the hyperoptimisation procedure.

Weighted fits: the user can choose to give additional weight to specific datasets when computing the total χ^2.

This feature can be useful to investigate in more detail the relative impact that such datasets have in the global fit, and explore possible tensions with other datasets or groups of processes following the strategy laid out in [5].

The run cards required for producing the main NNPDF4.0 fits are stored under

https://github.com/NNPDF/nnpdf/tree/master/n3fit/runcards/reproduce_nnpdf40/

These enable users to readily reproduce the results and also generate modifications of dataset selection, methodology or theory choices by suitably tweaking a run card.

Performance. One of the main advantages introduced by the new methodology underlying NNPDF4.0 in comparison to its predecessors using genetic algorithms is the significant fitting speed up achieved. As an illustration of this improvement in performance, we note that the NNPDF4.0 NNLO global fit takes fewer than 6 hours per replica on a single CPU core, as compared to $\simeq$ 36 hours using the NNPDF3.1-like methodology. This significant reduction of the CPU footprint of the global PDF fits leads to a faster production rate of fit variants, and it also allows one the prototyping of new approaches to PDF fitting using deep learning. Indeed, technologies such as hyperoptimisation were previously impractical but with the improved computational performance of the NNPDF code they are used in the fit. Furthermore, with the use of `TensorFlow` in the fitting toolkit, the ability to conveniently perform fits on the Graphics Processing Unit (GPU) might allow for further improvements in performance as suggested by the study in Ref. [40]. Such an implementation in the main `NNPDF` code is reserved for a future release.

3 The NNPDF analysis code: `validphys`

The `validphys` toolkit is at the heart of the NNPDF code base, bridging together the other components and providing basic data structures, compatibility interfaces, I/O operations and algorithms. These are used to assemble a suite of statistical analysis and plotting tools. We describe it here, and refer the reader to the publications mentioned in the code structure description in Sect. 2 as well as the online documentation of the NNPDF framework for further details on the other parts of the code.

The `validphys` code is in turn built on top `reportengine` [41], a data analysis framework which seeks to achieve the following goals:

- To aid structuring data science code bases so as to make them understandable and lower the entry barrier for new users and developers.
- To provide a declarative interface that allows the user specifying the required analysis by providing a minimal amount of information in the form of a run card, making the analysis reproducible given said run card.
- To provide a robust environment for the execution of data analysis pipelines including robust error checking, automatic documentation, command-line tools and interactive applications.

The key observation underpinning the design of `reportengine` is that most programming tasks in data science correspond to codes that are fully deterministic given their input. Every such program can be seen as a direct acyclic graph (DAG), see example the one shown in Fig. 2, with links representing the dependencies between a given step of the computation and the subsequent ones. Specifically, the nodes in such graph (resources) correspond to results of executing functions (providers) which usually correspond to functions in the PYTHON programming language. These functions are required to be pure, that is, such that their outputs are deterministic functions of the inputs and that no side effects that alter the state of the program happen.[1] These side effects are typically managed by the `reportengine` framework itself, with tools to, for example, save image files to a suitably unique filesystem location.

[1] Note that the concept of pure function is used here somewhat more loosely than in programming languages such as `Haskell` [42], since side effects such as logging information or writing files to disk are allowed as long as they are idempotent.

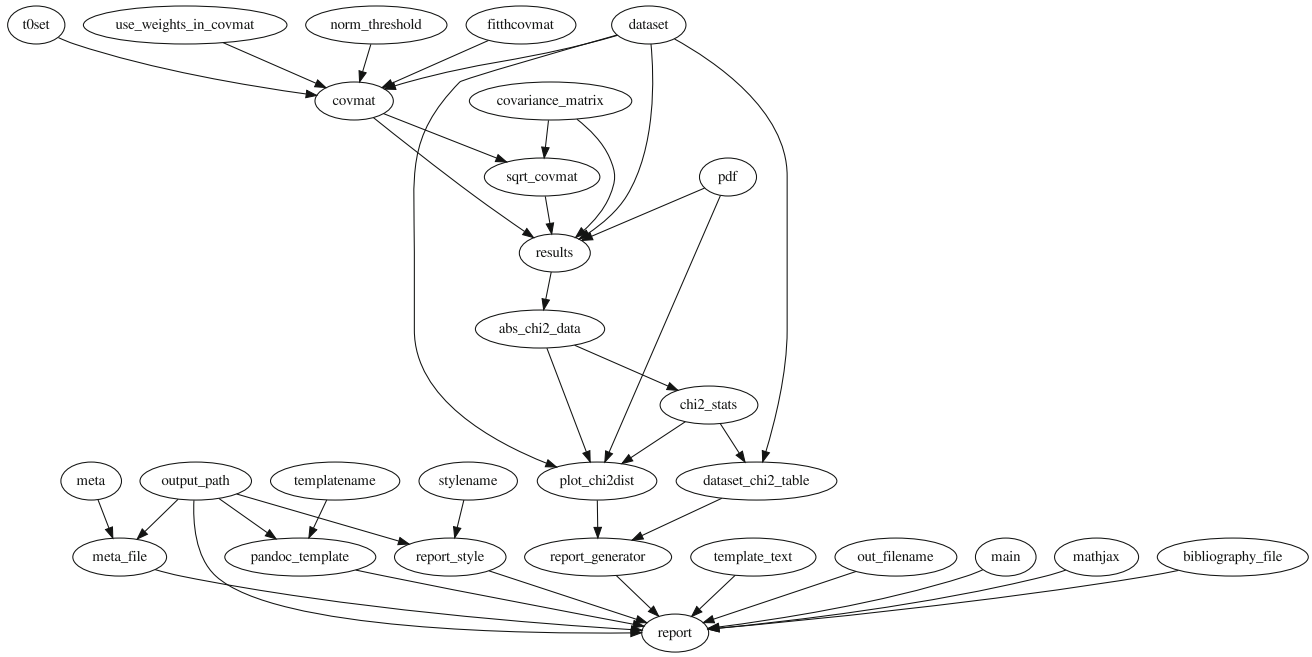

Fig. 2 Direct acyclic graph corresponding to the run card provided in Fig. 3. The graph shows the inputs extracted from the run card, such as `pdf` (the PDF set) and the `dataset`, the intermediate steps required for the χ^2 computation (such as evaluating the `covariance_matrix`), and the final target requested by the user, in this case a training `report` containing a histogram and a table with the χ^2 values obtained for this dataset for the indicated input PDF and choice of theory settings

The goal of simplifying the programming structure is achieved then by decomposing the program in terms of pure functions. Code developers are required to reason about the inputs of each individual function as well as its code, but not about any global state of the program or the order of execution, with the problem of putting the program together being delegated to the framework.

The `reportengine` framework has extensive facilities for automatically building the computation graph from the provided input. Users are only required to specify the ultimate target of the analysis (such as a figure, table, or report) with the intermediate steps being deduced thanks to a set of conventions in the program structure and a collection of utilities provided by the framework (for example tools to implement the map-reduce pattern). This allows complex analyses to be specified by purely declarative run cards without the need to write custom code for each of them. In turn, the run cards allow any user to precisely reproduce the results based on it and the corresponding version of the code.

A simple `validphys` run card, illustrating a minimal analysis of a dataset is shown in Fig. 3 with the DAG it spawns in Fig. 2.

As an example of the meta-programming features of `reportengine`, the `template_text` input in the run-card displayed in Fig. 3 illustrates how it is possible to spawn arbitrary other actions, with their corresponding dependencies, based on the user input as shown in Fig. 2. The framework allows implementing similar complex workflows with its programming interface. Users are referred to the online documentation for further details, code references, and specific examples.

The introspection capabilities of `reportengine` enable it to provide a robust and convenient environment for carrying out analysis. Most notably they enable specifying powerful checks on the user input. Basic constraints are implemented by instrumenting type annotations of PYTHON functions, which are used to verify that data types in the run cards match those expected by the code, but in addition arbitrary checks can also be attached to both input values or provider functions. This is commonly known as contract programming, but differs with many implementations in that checks are executed at the time the DAG is being built instead of when functions are executed. Thus, the DAG construction phase can be seen as a compilation phase, where developers have the ability to write arbitrary compiler checks. This feature allows eliminating large classes of runtime errors, thereby increasing the chances that the analysis runs to completion once the DAG has been constructed and checked. Another introspection feature consists of the capability of tracing the required inputs for a given provider and displaying them as automatically generated command line documentation.

As an implementation of the `reportengine` framework the `validphys` code features workflow focused on

```
dataset_input:
    dataset:  ATLAS_WP_JET_8TEV_PT
    cfac: [QCD]

theoryid: 200

use_cuts: "nocuts"

pdf: NNPDF31_nnlo_as_0118

template_text: |
    # Histogram

    {@plot_chi2dist@}

    # Table

    {@dataset_chi2_table@}


actions_:
    - report(main=True)
```

Fig. 3 A `validphys` runcard which produces a report containing a table and a histogram with the χ^2 values obtained for the ATLAS $W^+ +$ jets 8 TeV differential distributions when using the $N_{\rm rep} = 100$ replicas of NNPDF3.1 NNLO as input dataset and the theory settings specified by the `theoryid: 200` of the database. In particular, the runcard specifies the string for the `dataset`, the use of QCD K-factors, and the requirements that no kinematic cuts should be applied to the input dataset. Possible input options are referenced in Sec. 2. The DAG graph corresponding to the execution of this runcard is represented in Fig. 2

declarative and reproducible run cards. The code relies on common `Python` data science libraries such as `NumPy` [43], `SciPy` [44], `Matplotlib` [45] and `Pandas` [46] through its use of `Pandoc` [47], and it implements data structures that can interact with those of `libnnpdf`, as well as with analogs written in pure PYTHON. These include NNPDF fits, `LHAPDF` grids, and `FK`-tables. In addition, the code allows to quickly acquire relevant data and theory inputs by automatically downloading them from remote sources whenever they are required in a runcard. It also contains tooling to upload analysis results to an online server, to share it with other users or developers, and to allow it to be reproduced by other parties.

Some common data analysis actions that can be realised within the `validphys` framework include:

- Evaluating the convolutions between `FK`-tables and PDF sets, to evaluate in a highly efficient manner the theoretical predictions for the cross-sections of those datasets and theory settings we have implemented.
  Note that here any input PDF set can be used, not only NNPDF sets.
- Producing data versus theory comparison plots allowing for the graphical visualisation of the wealth of experimental measurements implemented in the NNPDF framework matched against the theory predictions.
  Again, predictions for arbitrary PDF sets can be used as input.
- Computing statistical estimators based on such data versus theory comparison, such as the various types of χ^2 [22], together with many plotting and grouping options.
- A large variety of plotting tools and options for the PDFs and partonic luminosities, including plots in arbitrary PDF bases.
  Some of these functionalities are related to those provided by the `APFEL-Web` online PDF plotter [48].
- Manipulating LHAPDF grids, implementing operations such as Hessian conversions [49,50].

The typical output of `validphys` is an HTML report containing the results requested by the user via the runcard. Fig. 4 displays the report obtained after executing the runcard in Fig. 3, consistent of an histogram displaying the distribution of χ^2 values for the $N_{\rm rep} = 100$ replicas of the NNPDF3.1 NNLO set when its predictions based on the `theoryid:200` theory settings are compared to the ATLAS W, Z 13 TeV total cross-sections. In order to highlight the potential of `validphys`, we have collected in this link

https://data.nnpdf.science/nnpdf40-reports/

representative training reports corresponding to the NNPDF4.0 analysis, such as comparisons between fits at different perturbative orders and between fits based on different datasets.

Additional features of the current release of the `validphys` framework include tools that make possible:

- Comparing two PDF fits by means of the `vp-comparefits` tool, which generates a report composed by almost 2000 figures and and 12 tables, displaying fit quality estimators, PDF comparisons, data-theory comparisons and positivity observables.
- Carrying out and characterising closure tests [39] and future tests [52].
- Performing simultaneous fits of the PDFs together with the strong coupling constant [38].
- Evaluating the theory covariance matrix constructed from scale variations, which can then be used as input for PDF fits accounting for missing higher order uncertainties (MHOUs) following the strategy of [53,54].
- Studying Hessian PDF tolerances.
- Determining Wilson coefficients in the Effective Field Theory (EFT) framework together with PDFs following the strategy presented in [15,16].

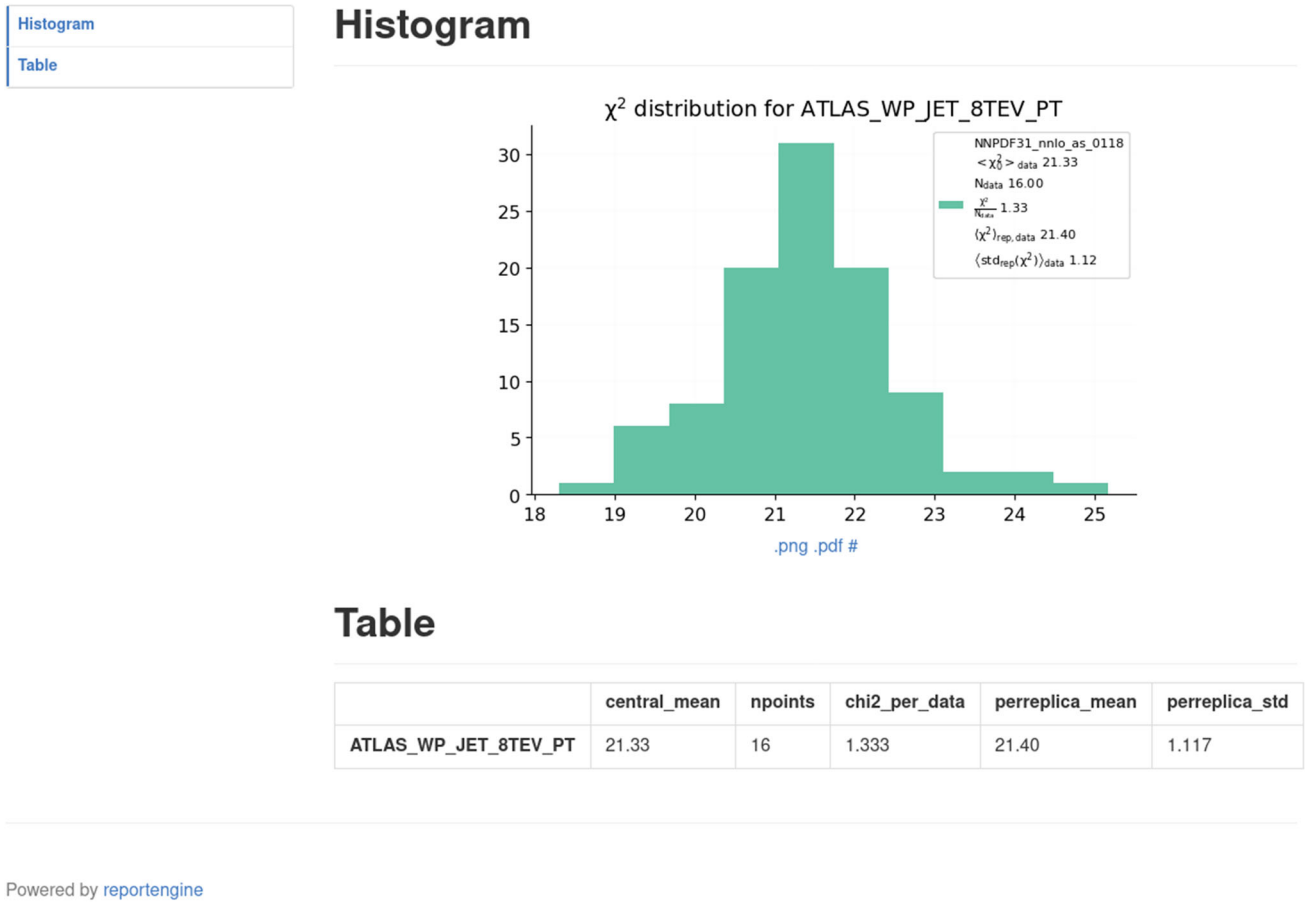

Fig. 4 The output of executing the runcard in Fig. 3 with `validphys` is an HTML report consistent of an histogram and the corresponding table indicating the distribution of χ^2 values over the $N_{\rm rep} = 100$ replicas of NNPDF3.1 NNLO for the ATLAS $W^+ +$ jets 8 TeV differential distributions [51] and the `theoryid:200` theory settings

– Analysing theoretical prediction with matched scale variations.

In conclusion, it is worth emphasising that many of the `validphys` features described here can be deployed outside the framework of the NNPDF fits. For instance, the tooling to evaluate the theory covariance matrix could also be relevant in the context of Hessian PDF fits, and comparisons between PDF sets can be carried out for other fits beyond NNPDF, provided one is careful and adopts consistent theoretical settings for each of the inputs.

4 Applications

Let us briefly discuss now some possible future applications of the NNPDF fitting framework presented in this work. As representative examples, we consider the inclusion of new experimental data, producing fits varying the theory settings, and going beyond the determination unpolarised collider PDFs. We discuss each of these applications in turn, and for a more comprehensive list we refer the interested user to the online documentation.

Adding new experimental data to the global fit. A typical application of the open-source NNPDF framework would be that of assessing the impact of some new measurement in the global PDF fit. Carrying out a full-fledged fit has several advantages as compared to approximate methods such as Bayesian reweighting [55,56], in particular one does not rely on the availability of prior fits composed by a very large number of replicas. Also, this way the user can easily vary the input dataset or the theoretical settings of this baseline fit. Furthermore, it is possible to add simultaneously to the global fit a large number of new datasets, while the reliability of the reweighting procedure is typically limited to a single dataset.

To implement a new dataset in the NNPDF code, one should start by adding the new measurement to the `buildmaster` suite. This will parse the new data points

into the common format suitable for its use in the PDF fits. Such an implementation will in general include information regarding the data central values, specification of the kinematic variables, statistical uncertainties and any relevant correlated systematic uncertainties that may exist in the dataset. In particular, the systematic uncertainties must be accompanied by metadata specifying their type (i.e if they are multiplicative or additive) as well as any possible correlations they may have with other systematic uncertainties (for example, the luminosity uncertainty will often be correlated across a given experiment). These uncertainties will then be used to construct the covariance matrix as well as the Monte Carlo replicas used to train the neural networks parametrising the PDFs.

Furthermore, in order to run the fit, the user would have to produce the corresponding `FK`-tables for this new dataset, which implies evaluating the fast NLO grids via `APPLgrid`, `FastNLO`, or `PineAPPL` [57] and then combining them with the DGLAP evolution kernels via `APFELcomb`. Depending on the perturbative order and electroweak settings of the fit, one needs to complement this `FK`-table with bin-by-bin NNLO QCD and/or NLO electroweak K-factors. With these ingredients, it is then be possible to add the data to a NNPDF fit and gauge its impact by comparing to a baseline with this same dataset excluded. If the impact of the dataset on the PDFs is moderate one can adopt the same hyperparameters as in the baseline reference; however, it is recommended practice to verify the stability of the fit results with respect a dedicated round of hyperoptimisation. Note also that new theory constraints, e.g. as those that could be imposed by lattice QCD calculations, can be accounted for in the same manner is with the addition of a new dataset.

As a consequence of adding the new dataset to those already packaged within the NNPDF code, the user now has access to the `validphys` tools described in Sect. 3, and hence they can easily characterise the goodness of the fit and quantify the agreement with the theory predictions, and well as assess the impact of this new dataset into the PDFs, partonic luminosities, and physical cross-sections.

NNPDF fits with different theory settings. Another foreseeable application of the open source fitting code is to produce variants of the NNPDF global analyses with modifying settings for the theoretical calculations. For instance, in determinations of the strong coupling $\alpha_s(m_Z)$ from collider data, one typically needs a series of PDF fits with a wide range and fine spacing of α_s values. These dedicated fits can be produced with the NNPDF code, and in addition while producing such PDF fits the user can also choose to tune the input dataset, e.g. by excluding specific types of processes, and the theory settings, e.g. with different variable-flavour-number-scheme. As emphasised in [58], when extracting SM parameters such as $\alpha_s(m_Z)$ from datasets sensitive to PDFs, it it necessary to simultaneously account for the impact of such datasets on the PDFs themselves (and not only on α_s) to avoid biasing the determination. Hence, these varying-α_s fits should already include the dataset from which α_s will be extracted, and this is only possible thanks to the availability of the NNPDF open source code.

The same caveats apply in the case of determinations of the heavy quark (charm, bottom, and top) masses from collider processes in which PDFs also enter the theory calculations. Other possible examples of NNPDF fits with varying theory settings are fits with different flavour assumptions, DGLAP evolution settings, or with approximations for unknown higher order perturbative corrections such as those evaluated from resummation. One may also be interested in tailored PDF sets for specific cross-section calculations, such as the doped PDFs [59] where the running with the active number of flavours n_f is different for $\alpha_s(Q)$ and for the PDF evolution.

In order to run a variant of the NNPDF fit with different theory settings, the user needs to verify if the corresponding sought-for `theory-id` already exists in the `theory` database. If this is the case, the fit with the new theory settings can be easily produced by adjusting the `theory-id` parameter in the run card. If, however, the `FK`-tables with the required theory settings are not available in the database, the user needs first to produce them using `APFELcomb`. We note that this is a relatively inexpensive step from the computational point of view, provided the corresponding NLO fast grids and the associated K-factors have been already produced. The user can follow the instructions in

https://docs.nnpdf.science/tutorials/apfelcomb.html

to produce `FK`-tables with their desired settings and assign them to a new `theory-id` in the theory database. By means of the `validphys` tooling, this new set of `FK`-tables can also be uploaded to the theory server where it will become available for other users.

Beyond unpolarised collinear PDFs. The current version of the NNPDF code focuses on unpolarised parton distributions. However, its flexible and modular infrastructure can be extended to the determination of related non-perturbative QCD quantities by means of the same methodology. While the NNPDF approach has also been used for the determination of polarised PDFs [60,61], fragmentation functions [62,63], and nuclear PDFs [64,65], in all these cases the code infrastructure only partially overlaps with that underlying NNPDF4.0. For instance, the polarised PDF determination rely on the FORTRAN predecessor of the NNPDF code, while the nuclear PDF fits adopt the `FK`-table approach for theoretical calculations but are based on a stand-alone machine learning framework. The availability of the NNPDF framework as open source code should hence lead to progress into

its extension to other quantities beyond unpolarised collinear PDFs, as well as for the determination of the collinear PDFs of different hadronic species such as pions or kaons. These studies are especially interesting at the light of future experiments with focus on testing the nucleon, nuclear, and mesonic structure, from the Electron Ion Colliders [66,67] to AMBER at the CERN-SPS [18].

A closely related application of the NNPDF fitting code would be the simultaneous determination of non-perturbative QCD quantities exhibiting non-trivial cross-talk, such as nucleon and nuclear PDFs [68], (un)polarised PDFs together with fragmentation functions [69], or collinear and transverse-momentum-dependent PDFs. Such integrated global PDF determinations have many attractive features, for instance in the proton global analysis it would not be necessary anymore to treat in a special manner the deuteron and heavy nuclear datasets (since the A dependence would be directly extracted from the data), and the interpretation of processes such as semi-inclusive DIS (SIDIS) would not rely on assumptions about the behaviour of either the nucleon PDFs (for the initial state) or the fragmentation functions (for the final state). Clearly, a pre-requisite for such integrated fits is the availability of the code infrastructure for the determination of the individual non-perturbative QCD quantities within the public NNPDF framework.

5 Conclusions

In this work we have presented the public release, as an open-source code, of the software framework underlying the recent NNPDF4.0 global determination of parton distributions. The flexible and robust NNPDF code exploits state-of-the-art developments in machine learning to realise a comprehensive determination of the proton structure from a wealth of experimental data. The availability of this framework as open source should encourage the broader high-energy and nuclear physics communities to deploy machine learning methods in the context of PDF studies.

Among the wide range of possible user cases provided by the NNPDF code, one can list assessing the impact of new data, producing tailored fits with variations of SM parameters such as $\alpha_s(m_Z)$ or m_c for their simultaneous extraction together with the PDFs, and studying the eventual presence of beyond the SM physics in precision LHC measurements of the high-p_T tails of kinematic distributions using effective field theories. Furthermore, the determination of related non-perturbative QCD quantities from nuclear PDFs and polarised PDFs to fragmentation functions represents another potential application of the NNPDF framework

In order facilitate these various applications, the NNPDF codebase is now almost entirely written in PYTHON, the currently *de facto* standard choice of programming language within both the data science as well as the scientific community. With the majority of the libraries being highly efficient wrappers of faster languages, PYTHON is no longer bottlenecked by performance and so its relatively low barrier of entry should allow for the NNPDF code to be modified and expanded. With this motivation, we have discussed how the user may wish to configure a run card for their PDF fit, indicated the details of the parameters that are exposed to the user, and presented the `validphys` library which acts as an in-house analysis suite designed to be not only reproducible, but also allowing for complex tasks to be achieved using transparent run card based inputs.

We reiterate that we have restricted ourselves to a succinct high-level summary of the main functionalities of the NNPDF code. The main reference for the interested user is online documentation which accompanies this release, which features technical commentary as well as example use cases. The documentation is kept continuously up-to-date following the ongoing development of the code.

Acknowledgements S. C., S. F., J. C.-M., R. S. and C. S. are supported by the European Research Council under the European Union's Horizon 2020 research and innovation Programme (grant agreement n.740006). M. U. and Z. K. are supported by the European Research Council under the European Union's Horizon 2020 research and innovation Programme (grant agreement n.950246). M. U. and S. I. are partially supported by the Royal Society grant RGF/EA/180148. The work of M. U. is also funded by the Royal Society grant DH150088. The work of M. U., S. I., C. V. and Z. K. is partially supported by the STFC consolidated grant ST/L000385/1. The work of Z. K. was partly supported by supported by the European Research Council Consolidator Grant "NNLOforLHC2". J. R. is partially supported by NWO, the Dutch Research Council. C. V. is supported by the STFC grant ST/R504671/1. T. G. is supported by The Scottish Funding Council, grant H14027. R. L. P. and M. W. by the STFC grant ST/R504737/1. R. D. B., L. D. D. and E. R. N. are supported by the STFC grant ST/P000630/1. E. R. N. was also supported by the European Commission through the Marie Sklodowska-Curie Action ParDHonS (grant number 752748).

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: All data are publicly released on the github repository https://github.com/NNPDF and the online documentation is released at https://docs.nnpdf.science//.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
Funded by SCOAP[3].

References

1. Physics of the HL-LHC Working Group Collaboration, M. Cepeda et al., Higgs physics at the HL-LHC and HE-LHC. arXiv:1902.00134
2. P. Azzi et al., Report from Working Group 1: standard model physics at the HL-LHC and HE-LHC. CERN Yellow Rep. Monogr. **7**, 1–220 (2019). arXiv:1902.04070
3. J. Gao, L. Harland-Lang, J. Rojo, The structure of the proton in the LHC precision era. Phys. Rep. **742**, 1–121 (2018). arXiv:1709.04922
4. J.J. Ethier, E.R. Nocera, Parton distributions in nucleons and nuclei. Annu. Rev. Nucl. Part. Sci. **70**, 43–76 (2020). arXiv:2001.07722
5. NNPDF Collaboration, R.D. Ball et al., The path to proton structure at one-percent accuracy. arXiv:2109.02653
6. NNPDF Collaboration, R.D. Ball et al., Parton distributions from high-precision collider data. Eur. Phys. J. C **77**(10), 663. arXiv:1706.00428 (2017)
7. A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page et al., LHAPDF6: parton density access in the LHC precision era. Eur. Phys. J. C **75**, 132 (2015). arXiv:1412.7420
8. J. Butterworth et al., PDF4LHC recommendations for LHC Run II. J. Phys. **G43**, 023001 (2016). arXiv:1510.03865
9. J. Rojo et al., The PDF4LHC report on PDFs and LHC data: Results from Run I and preparation for Run II. J. Phys. G **42**, 103103 (2015). arXiv:1507.00556
10. M.D. Wilkinson et al., The FAIR guiding principles for scientific data management and stewardship. Sci. Data **3**, 160018 (2016)
11. NNPDF Collaboration, R.D. Ball et al., Nnpdf/nnpdf: nnpdf v4.0.3. https://doi.org/10.5281/zenodo.5362228 (2021)
12. R.D. Ball, E.R. Nocera, J. Rojo, The asymptotic behaviour of parton distributions at small and large x. Eur. Phys. J. **C76**(7), 383 (2016). arXiv:1604.00024
13. H.-W. Lin et al., Parton distributions and lattice QCD calculations: a community white paper. Prog. Part. Nucl. Phys. **100**, 107–160 (2018). arXiv:1711.07916
14. K. Cichy, L. Del Debbio, T. Giani, Parton distributions from lattice data: the nonsinglet case. JHEP **10**, 137 (2019). arXiv:1907.06037
15. S. Carrazza, C. Degrande, S. Iranipour, J. Rojo, M. Ubiali, Can new physics hide inside the proton? Phys. Rev. Lett. **123**(13), 132001 (2019). arXiv:1905.05215
16. A. Greljo, S. Iranipour, Z. Kassabov, M. Madigan, J. Moore, J. Rojo, M. Ubiali, C. Voisey, Parton distributions in the SMEFT from high-energy Drell–Yan tails. arXiv:2104.02723
17. E.L. Berger, M. Guzzi, H.-L. Lai, P.M. Nadolsky, F.I. Olness, Constraints on color-octet fermions from a global parton distribution analysis. Phys. Rev. D **82**, 114023 (2010). arXiv:1010.4315
18. B. Adams et al., Letter of intent: a new QCD facility at the M2 beam line of the CERN SPS (COMPASS++/AMBER). arXiv:1808.00848
19. S. Alekhin et al., HERAFitter. Eur. Phys. J. **C75**(7), 304 (2015). arXiv:1410.4412
20. xFitter Team Collaboration, O. Zenaiev, xFitter project. PoS DIS2016, 033 (2016)
21. E. Maguire, L. Heinrich, G. Watt, HEPData: a repository for high energy physics data. J. Phys. Conf. Ser. **898**(10), 102006 (2017). arXiv:1704.05473
22. The NNPDF Collaboration, R.D. Ball et al., Fitting parton distribution data with multiplicative normalization uncertainties. JHEP **05**, 075 (2010). arXiv:0912.2276
23. R.D. Ball, S. Carrazza, L. Del Debbio, S. Forte, J. Gao et al., Parton distribution benchmarking with LHC data. JHEP **1304**, 125 (2013). arXiv:1211.5142
24. T. Carli et al., A posteriori inclusion of parton density functions in NLO QCD final-state calculations at hadron colliders: the APPLGRID Project. Eur. Phys. J. C **66**, 503 (2010). arXiv:0911.2985
25. fastNLO Collaboration, M. Wobisch, D. Britzger, T. Kluge, K. Rabbertz, F. Stober, Theory-data comparisons for jet measurements in hadron-induced processes. arXiv:1109.1310
26. V. Bertone, S. Carrazza, J. Rojo, APFEL: a PDF evolution library with QED corrections. Comput. Phys. Commun. **185**, 1647 (2014). arXiv:1310.1394
27. V. Bertone, S. Carrazza, N.P. Hartland, APFELgrid: a high performance tool for parton density determinations. Comput. Phys. Commun. **212**, 205–209 (2017). arXiv:1605.02070
28. Stefano Carrazza, Juan Cruz-Martinez, Towards a new generation of parton densities with deep learning models. Eur. Phys. J. C **79**(8), 676 (2019)
29. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for large-scale machine learning, in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({ OSDI} 16)*, pp. 265–283 (2016)
30. J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D.D. Cox, Hyperopt: a Python library for model selection and hyperparameter optimization. Comput. Sci. Discov. **8**, 014008 (2015)
31. A. Candido, S. Forte, F. Hekhorn, Can $\overline{\text{MS}}$ parton distributions be negative? JHEP **11**, 129 (2020). arXiv:2006.07377
32. NNPDF Collaboration, R.D. Ball et al., Parton distributions for the LHC Run II. JHEP **04**, 040 (2015). arXiv:1410.8849
33. NNPDF Collaboration, R.D. Ball, V. Bertone, M. Bonvini, S. Carrazza, S. Forte, A. Guffanti, N.P. Hartland, J. Rojo, L. Rottoli, A determination of the charm content of the proton. Eur. Phys. J. C **76**(11), 647 (2016). arXiv:1605.06515
34. NNPDF Collaboration, 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), 008 (2018). arXiv:1712.07053
35. R.D. Ball, V. Bertone, M. Bonvini, S. Marzani, J. Rojo, L. Rottoli, Parton distributions with small-x resummation: evidence for BFKL dynamics in HERA data. Eur. Phys. J. **C78**(4), 321 (2018). arXiv:1710.05935
36. D.M. Beazley, SWIG, in *An easy to use tool for integrating scripting languages with c and c++*, vol. TCLTK'96, p. 15 (USENIX Association, 1996)
37. J. Gao, Massive charged-current coefficient functions in deep-inelastic scattering at NNLO and impact on strange-quark distributions. JHEP **02**, 026 (2018). arXiv:1710.04258
38. NNPDF Collaboration, R.D. Ball, S. Carrazza, L. Del Debbio, S. Forte, Z. Kassabov, J. Rojo, E. Slade, and M. Ubiali, Precision determination of the strong coupling constant within a global PDF analysis. Eur. Phys. J. C **78**(5), 408 (2018). arXiv:1802.03398
39. NNPDF Collaboration, Bayesian approach to inverse problems: an application to NNPDF closure testing **(in preparation)**
40. S. Carrazza, J.M. Cruz-Martinez, M. Rossi, PDFFlow: parton distribution functions on GPU. Comput. Phys. Commun. **264**, 107995 (2021). arXiv:2009.06635
41. Z. Kassabov, Reportengine: a framework for declarative data analysis. https://doi.org/10.5281/zenodo.2571601 (2019)
42. A. Mena, *Practical Haskell: A Real World Guide to Programming* (Apress, 2019)
43. C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy. Nature **585**, 357–362 (2020)
44. P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov,

A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P., van Mulbregt, and SciPy 1.0 contributors, SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods **17**, 261–272 (2020)
45. J.D. Hunter, Matplotlib: a 2d graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007)
46. W. McKinney, Data structures for statistical computing in Python. in *Proceedings of the 9th Python in Science Conference* (S. van der Walt, J. Millman, eds.), pp. 56 – 61 (2010)
47. J. MacFarlane, *Pandoc: a universal document converter*. http://pandoc.org (2013)
48. S. Carrazza, A. Ferrara, D. Palazzo, J. Rojo, APFEL Web: a web-based application for the graphical visualization of parton distribution functions. J. Phys. G **42**, 057001 (2015). arXiv:1410.5456
49. S. Carrazza, S. Forte, Z. Kassabov, J.I. Latorre, J. Rojo, An unbiased Hessian representation for Monte Carlo PDFs. Eur. Phys. J. C **75**(8), 369 (2015). arXiv:1505.06736
50. S. Carrazza, S. Forte, Z. Kassabov, J. Rojo, Specialized minimal PDFs for optimized LHC calculations. Eur. Phys. J. C **76**(4), 205 (2016). arXiv:1602.00005
51. ATLAS Collaboration, M. Aaboud et al., Measurement of differential cross sections and W^+/W^- cross-section ratios for W boson production in association with jets at $\sqrt{s} = 8$ TeV with the ATLAS detector. JHEP **05**, 077 (2018). arXiv:1711.03296 [Erratum: JHEP 10, 048 (2020)]
52. J. Cruz-Martinez, S. Forte, E.R. Nocera, Future tests of parton distributions. Acta Phys. Polon. B **52**, 243 (2021). arXiv:2103.08606
53. NNPDF Collaboration, R. Abdul Khalek et al., Parton distributions with theory uncertainties: general formalism and first phenomenological studies. Eur. Phys. J. C **79**(11), 931 (2019). arXiv:1906.10698
54. NNPDF Collaboration, R. Abdul Khalek et al., A first determination of parton distributions with theoretical uncertainties. Eur. Phys. J. C **79**, 838 (2019). arXiv:1905.04311
55. R.D. Ball, V. Bertone, F. Cerutti, L. Del Debbio, S. Forte et al., Reweighting and unweighting of parton distributions and the LHC W lepton asymmetry data. Nucl. Phys. B **855**, 608–638 (2012). arXiv:1108.1758
56. The NNPDF Collaboration, R.D. Ball et al., Reweighting NNPDFs: the W lepton asymmetry. Nucl. Phys. B **849**, 112–143 (2011). arXiv:1012.0836
57. S. Carrazza, E.R. Nocera, C. Schwan, M. Zaro, PineAPPL: combining EW and QCD corrections for fast evaluation of LHC processes. JHEP **12**, 108 (2020). arXiv:2008.12789
58. 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), 182 (2020). arXiv:2001.04986
59. V. Bertone, S. Carrazza, J. Rojo, Doped parton distributions, in *27th Rencontres de Blois on Particle Physics and Cosmology*, 9 (2015). arXiv:1509.04022
60. NNPDF Collaboration, E.R. Nocera, R.D. Ball, S. Forte, G. Ridolfi, J. Rojo, A first unbiased global determination of polarized PDFs and their uncertainties. Nucl. Phys. B **887**, 276 (2014). arXiv:1406.5539
61. The NNPDF Collaboration, R.D. Ball et al., Unbiased determination of polarized parton distributions and their uncertainties. Nucl. Phys. B **874** 36–84 (2013). arXiv:1303.7236
62. NNPDF Collaboration, V. Bertone, S. Carrazza, N.P. Hartland, E.R. Nocera, J. Rojo, A determination of the fragmentation functions of pions, kaons, and protons with faithful uncertainties. Eur. Phys. J. C **77**(8), 516 (2017). arXiv:1706.07049
63. NNPDF Collaboration, V. Bertone, N.P. Hartland, E.R. Nocera, J. Rojo, L. Rottoli, Charged hadron fragmentation functions from collider data. Eur. Phys. J. C **78**(8), 651 (2018). arXiv:1807.03310
64. NNPDF Collaboration, R. Abdul Khalek, J.J. Ethier, J. Rojo, Nuclear parton distributions from lepton-nucleus scattering and the impact of an electron-ion collider. Eur. Phys. J. C **79**(6), 471 (2019). arXiv:1904.00018
65. R. Abdul Khalek, J.J. Ethier, J. Rojo, G., van Weelden, nNNPDF2.0: quark flavor separation in nuclei from LHC data. JHEP **09**, 183 (2020). arXiv:2006.14629
66. D.P. Anderle et al., Electron-ion collider in China. Front. Phys. (Beijing) **16**(6), 64701 (2021). arXiv:2102.09222
67. R. Abdul Khalek et al., Science requirements and detector concepts for the electron-ion collider: EIC Yellow Report. arXiv:2103.05419
68. R.A. Khalek, J.J. Ethier, E.R. Nocera, J. Rojo, Self-consistent determination of proton and nuclear PDFs at the Electron Ion Collider. Phys. Rev. D **103**(9), 096005 (2021). arXiv:2102.00018
69. Jefferson Lab Angular. Momentum, (JAM) Collaboration, E. Moffat, W. Melnitchouk, T. C. Rogers, N. Sato, Simultaneous Monte Carlo analysis of parton densities and fragmentation functions. Phys. Rev. D **104**(1), 016015. arXiv:2101.04664 (2021)