Novel parton density determination code

Francesca Capel,[1,*] Ritu Aggarwal,[2,†] Michiel Botje,[3,‡] Allen Caldwell,[1,§]
Oliver Schulz,[1,‖] and Andrii Verbytskyi[1,¶]

[1] *Max-Planck-Institut für Physik, München, Germany*
[2] *USAR, Guru Gobind Singh Indraprastha University, East Delhi-110032, India*
[3] *Nikhef, Amsterdam, The Netherlands*

(Received 1 February 2024; accepted 6 June 2024; published 18 July 2024)

We introduce our novel Bayesian parton density determination code, PartonDensity.jl. The motivation for this new code, the framework, and its validation are described. As we show, PartonDensity.jl provides both a flexible environment for the determination of parton densities and a wealth of information concerning the knowledge update provided by the analyzed dataset.

DOI: 10.1103/PhysRevD.110.014024

I. INTRODUCTION

This paper describes a novel approach to the determination of parton density functions (PDFs) of hadrons and presents the PartonDensity.jl programming code that implements this approach.

Our analysis method significantly differs from those pursued by other groups [1–10], allowing us to tackle datasets that have not been included in PDF determinations so far. There are two distinct features that set our code apart, as described in the following.

First of all, we use Bayesian techniques to determine the parton density functions.[1] This formulation allows us to include in a coherent and transparent way known constraints and results from previous theoretical and experimental analyses in prior probability distributions. The result of a Bayesian analysis is a multivariate posterior probability distribution of all model parameters, including nuisance parameters used to describe systematic uncertainties in the data. Correlations between any subset of parameters can be studied, providing vastly more information than results obtained otherwise. Systematic error propagation is simply achieved by integrating the posterior over the nuisance parameters. In addition, the information content of the data is easily judged by comparing the posterior to the input priors.

Secondly, we have implemented a forward modeling approach, where event numbers in kinematic bins are predicted and compared to observed event counts, which are always bin-by-bin uncorrelated with known Poisson distribution. This is in contrast to the analyses of unfolded cross sections, where a Gaussian statistical hypothesis is implied and where the data are always correlated in a way that is often not known. Another advantage of forward modeling is that it can correctly handle the low statistics case with empty or sparsely populated bins.

In principle, the Bayesian and forward modeling approaches can be used independently of each other. The current version of our code allows for the analysis of inclusive $e^{\pm}p$ scattering data, made available in the form of binned event counts. An extension to allow for the Bayesian analysis of differential cross sections extracted at the QED-Born level is in development and will be reported separately. We note that it is also possible to use the forward modeling approach within any likelihood-based analysis framework, not just the Bayesian formalism described here.

The PartonDensity.jl software package[2] is implemented in the modern Julia [13] language and uses several Julia packages for the analysis, the most important of which is the Bayesian Analysis Toolkit (BAT.jl) [14]. We have published an analysis of the ZEUS data [15,16] using PartonDensity.jl in [17]. In this work, we present our analysis method in more detail and demonstrate its validity through the application to relevant simulated pseudodata, for which the true PDFs are known.

This paper is organized as follows. In Sec. II, we begin with a general introduction to the Bayesian analysis

[*] Contact author: capel@mpp.mpg.de
[†] Contact author: ritu.aggarwal1@gmail.com
[‡] Contact author: m.botje@nikhef.nl
[§] Contact author: caldwell@mpp.mpg.de
[‖] Contact author: oschulz@mpp.mpg.de
[¶] Contact author: andrii.verbytskyi@mpp.mpg.de
[1] For other efforts toward PDF determination from a Bayesian perspective, see, e.g., [11,12].

[2] Available at https://github.com/cescalara/PartonDensity.jl.

approach. In Sec. III, we describe our forward model, or how the e$^{\pm}$p binned event counts are computed from a set of parametrized input PDFs. In the development of the code, a large amount of effort was dedicated to speeding up the necessary calculations, and we briefly report on these in Sec. IV. In Sec. V, we discuss our PDF parametrizations and prior parameter constraints. Finally, validation tests on pseudodata and goodness-of-fit tests are described in Secs. VI and VII, respectively.

II. BAYESIAN ANALYSIS APPROACH

A. Introduction

We use a Bayesian approach to obtain the joint posterior distribution $p(\theta|D, M)$ of a set of model parameters θ, conditional to the dataset D being analyzed. It is also conditional to the particular model choices M such as, among others, the modeling of systematic uncertainties in the data and the specific form of the PDFs chosen.

Because it is not possible to obtain an analytic expression for the posterior, Monte Carlo techniques are used to create parameter samples that are distributed according to $p(\theta|D, M)$. Nuisance parameters that are needed in the modeling of the data but are of no interest are removed by integrating the posterior over these. In this way, systematic uncertainties are automatically propagated in a consistent fashion by integrating over the systematic parameters in the model.

The resulting samples then enable us to study single- and multi-dimensional marginal distributions of the parameters of interest and extract all kinds of statistical estimators like mean values, credibility intervals, and so forth.

The posterior parameter distribution is obtained from Bayes' theorem,

$$p(\theta|D, M) = \frac{p(D|\theta, M) p(\theta|M)}{p(D|M)}. \tag{1}$$

Here $p(\theta|M)$ is the prior distribution of the parameters θ, and $p(D|\theta, M)$ is the probability to observe the data D, given a particular set of parameter values θ. In Eq. (1) it is evaluated as a function of θ for fixed data D and is called the likelihood function (it is not a probability density), denoted as

$$\mathcal{L}_D(\theta) = p(D|\theta), \qquad D \text{ fixed.}$$

Here and in the following, we will always imply the model choice M, but for clarity, it will be dropped in the notation.

An important feature of Bayes' theorem is the fact that the support of the posterior can never exceed that of the prior. This makes it straightforward to impose hard constraints on allowed parameter values, such as the requirement that a parameter is positive definite.

The normalization of the posterior is given by the so-called evidence,

$$p(D) = \int \mathcal{L}_D(\theta)\, p(\theta)\, \mathrm{d}\theta. \tag{2}$$

The evidence is a scalar value that is often challenging to compute as it results from a multidimensional integral. However, as it is independent of the parameter vector θ, its value is not needed to study the relative credibility of parameter values. Sampling algorithms like Markov Chain Monte Carlo (MCMC) make it possible to draw samples distributed according to the posterior distribution based on the non-normalized posterior density

$$\tilde{p}(\theta|D) = \mathcal{L}_D(\theta)\, p(\theta). \tag{3}$$

This yields posterior samples that can be used to produce the desired marginal parameter distributions, which can always be normalized afterward.

B. Implementation of MCMC

In our experience, prior parameter distributions with hard constraints, like positive definite parameters that may assume values close to zero can negatively impact MCMC convergence and efficiency. Several of the parameters of our model fall into this category. Furthermore, the momentum sum rule (see below) introduces a strict correlation between the momentum parameters that restricts the support of the posterior density to a subspace of the full parameter space. Densities with such a support typically cause sampling algorithms to fail completely, as the posterior density is zero almost everywhere in the parameter space. The methods to circumvent this obstacle are described below.

We solve the problem of sampling from a complicated distribution by performing a change of variables with the aim of creating a parameter space that is straightforward to sample from. Thus, instead of sampling the density $\tilde{p}(\theta|D)$ directly, we introduce a suitable transformation $\theta = f(x)$ and sample the density

$$\tilde{p}(x|D) = \mathcal{L}_D[f(x)]\mathcal{N}(x), \tag{4}$$

where $\mathcal{N}(x)$ is a multivariate Normal distribution of the appropriate dimension. The posterior $\tilde{p}(x|D)$ is unbounded and has full support over its parameter space, and so is much easier to sample. Samples of the original parameters θ are then obtained by simply applying the parameter transformation f to generated samples of the parameters x in the alternate space.

The challenge lies in finding suitable transformations f. For univariate components of the prior distribution, their quantile functions and the cumulative distribution function of the normal distribution provide the necessary building blocks [14]. For the Dirichlet distribution that we use to satisfy the momentum sum rule in the prior (see Sec. III B), a suitable transformation is given in [18].

C. Marginalization and uncertainty propagation

In many model analyses one is interested not in the full posterior distribution but in the marginal probability distribution of only one or a few parameters. For example, the probability distribution of a single parameter θ_i is obtained from

$$p(\theta_i|D) = \int p(\theta|D)\mathrm{d}\theta_{j\neq i}.$$

In practice, this calculation is performed quite simply by histogramming the θ_i values from all posterior samples, ignoring the other parameters. From such a histogram, several estimators can be computed to report the results. Commonly used quantities are

(a) *Mode of θ_i*. The value θ_i^* where the marginalized posterior probability density has a maximum. Modes are usually computed for fully marginalized posteriors $p(\theta_i|D)$ (“marginal mode“), for the entire posterior (“global mode”), or for any number of parameters, with the rest marginalized. Note that the parameter values that maximize the full posterior distribution usually do not coincide with those that maximize marginalized distributions.

(b) *Mean of θ_i*. This is the expectation value,

$$\langle\theta_i\rangle = \int_{\theta_{\min}}^{\theta_{\max}} p(\theta_i|D)\theta_i\mathrm{d}\theta_i,$$

with the parameter bounds denoted by $[\theta_{\min}, \theta_{\max}]$.

(c) *Median of θ_i*. The value $\hat{\theta}_i$ that splits the probability content of $p(\theta_i|D)$ in two:

$$\int_{\theta_{\min}}^{\hat{\theta}_i} p(\theta_i|D)\mathrm{d}\theta_i = \int_{\hat{\theta}_i}^{\theta_{\max}} p(\theta_i|D)\mathrm{d}\theta_i = 0.5.$$

(d) *Central interval*. The $(1-2\alpha)$ central interval $[\theta_-, \theta_+]$ is defined such that a fraction α of the probability is contained on either side of the interval:

$$\int_{\theta_{\min}}^{\theta_-} p(\theta_i|D)\mathrm{d}\theta_i = \int_{\theta_+}^{\theta_{\max}} p(\theta_i|D)\mathrm{d}\theta_i = \alpha.$$

(e) *Smallest interval*. The α smallest interval(s) is defined such that a fraction α of the probability is contained in a set of intervals where the set size is minimized. This is realized as a Lebesgue integral:

$$\int_{p(\theta_i|D)\geq p_{\min}} p(\theta_i|D)\mathrm{d}\theta_i = \alpha,$$

where $p_{\min}$ is to be determined. This procedure can result in several intervals in θ_i in the case of multimodal distributions.

(f) *Uncertainty propagation*. Having full access to the posterior allows for the evaluation of the probability distribution of any function of the parameters. In contrast to standard techniques used for error propagation, there is no need here for any assumptions like distributions being Gaussian shaped. As an example, consider that we have a function $f(x|\theta)$ of interest, for instance, a parton distribution that depends on a subset of the θ parameters. The distribution of $f(x)$ is then given by

$$p(f(x)|D) = \int p(f(x|\theta))p(\theta|D)\mathrm{d}\theta,$$

which can be evaluated in a straightforward way from the posterior samples. We give examples of such uses below.

D. BAT.jl—The Bayesian Analysis Toolkit

For the Bayesian inference process, we use the software package BAT.jl [14], which is a high-performance toolkit for Bayesian analysis tasks, coded in the Julia programming language [13]. The package provides multiple algorithms for posterior sampling, integration, and mode finding, as well as automatic plotting and reporting functionality.

The BAT.jl package also has the ability to automatically generate suitable space transformations between user-defined probability distributions and standard multivariate normal or uniform distributions. This enables us to automatically perform prior-space transformations as described in Sec. II B and sample the posterior via the Metropolis-Hastings algorithm in an unconstrained space where the prior has become a normal distribution in all dimensions. The samples are then automatically transformed back into the original space.

A full description of the algorithms and tuning of the BAT.jl toolkit can be found in [14].

III. ANALYSIS STRUCTURE

As mentioned in the Introduction, we focus in this initial release of our PDF determination code on the analysis of electron(positron)-proton deep inelastic scattering (DIS) data. The data are generally reported in bins of the scaling variables x and Q^2, which are computed from the reconstructed four momenta of the incoming proton and the incoming and scattered leptons (for more details see, e.g., [16]). In this section, we describe how the event numbers observed in these data are predicted, starting from a set of proton PDFs parametrized at some input scale Q_0^2.

A. Computation of the e$^{\pm}$p cross sections

To compute the e$^{\pm}$p deep inelastic cross sections, we start from a set of quark, antiquark, and gluon distributions $xf_i(x)$, parametrized at a fixed value of Q_0^2. The aim of the analysis is to determine from the data the posterior joint distribution of these parameters.

In the above, $f_i(x)$ is the number density of partons of type i in the proton, and $xf_i(x)$ is the momentum density of these partons.

The next step is to evolve these distributions in perturbative QCD [19–23] from the input scale to larger values of Q^2. We use the QCDNUM program [24] to evolve the PDFs in the Modified Minimal Subtraction ($\overline{\text{MS}}$)-scheme at next-to-next-to-leading order (NNLO) [25–34].

In the analysis, we impose the momentum sum rule and the valence quark counting rules. The momentum sum rule states that the fractional momenta of all partons in the proton add up to unity:

$$\sum_i \int_0^1 xf_i(x)\mathrm{d}x = \sum_i \Delta_i = 1. \qquad (5)$$

We introduce here the notation Δ_i for the total momentum fraction carried by the parton species i.

The quark counting rules fix the net number of quarks in the proton so that its quantum numbers are conserved:

$$\int_0^1 [q_i(x) - \bar{q}_i(x)]\mathrm{d}x = \begin{cases} 2 & \text{for } i = u, \\ 1 & \text{for } i = d, \\ 0 & \text{for } i = s, c, b, t. \end{cases} \qquad (6)$$

Here and in the following we use the notation q, $\bar{q}$, and g to denote quark, antiquark, and gluon densities.

It is important to point out that the QCD evolution equations guarantee that sum rules that are imposed at the starting scale Q_0^2 will be satisfied at all scales.

The neutral current DIS cross section for e$^{\pm}$p scattering is given in terms of generalized structure functions of the proton as (y is the inelasticity variable, see Ref. [16])

$$\frac{\mathrm{d}^2\sigma_{\mathrm{NC}}^{\mathrm{e}^{\pm}\mathrm{p}}}{\mathrm{d}x\mathrm{d}Q^2} = \frac{2\pi\alpha}{xQ^4}(Y_+\tilde{F}_2 \mp Y_- x\tilde{F}_3 - y^2\tilde{F}_{\mathrm{L}}), \qquad (7)$$

where $Y_{\pm} = 1 \pm (1-y)^2$, and α is the fine-structure constant.

The generalized structure functions are related to the vector and axial-vector coupling constants v_{e} and a_{e} by

$$\tilde{F}_i = F_i^{\gamma} - k_{\mathrm{Z}} v_{\mathrm{e}} F_i^{\gamma\mathrm{Z}} + k_{\mathrm{Z}}^2(v_{\mathrm{e}}^2 + a_{\mathrm{e}}^2)F_i^{\mathrm{Z}}, \qquad i = 2, \mathrm{L}$$
$$x\tilde{F}_3 = k_{\mathrm{Z}} a_{\mathrm{e}} xF_3^{\gamma\mathrm{Z}} + k_{\mathrm{Z}}^2 2v_{\mathrm{e}}a_{\mathrm{e}} xF_3^{\mathrm{Z}}, \qquad (8)$$

where

$$k_{\mathrm{Z}}(Q^2) = \frac{Q^2}{(Q^2 + m_{\mathrm{Z}}^2)4\sin^2\theta_{\mathrm{w}}\cos^2\theta_{\mathrm{w}}}.$$

In our analysis, we use $\sin^2\theta_{\mathrm{w}} = 0.231$ and $m_{\mathrm{Z}} = 91.1876$ GeV for the electroweak mixing angle and the Z-boson mass, respectively [35].

In leading order (LO) QCD, the structure functions are linear combinations of parton densities of different flavor (note that $F_{\mathrm{L}} = 0$ at LO):

$$\{F_2^{\gamma}, F_2^{\gamma\mathrm{Z}}, F_2^{\mathrm{Z}}\} = x\sum_i \{e_i^2, 2e_i g_{\mathrm{V}}, g_{\mathrm{A}}^2\}(q_i + \bar{q}_i)$$
$$x\{F_3^{\gamma}, F_3^{\gamma\mathrm{Z}}, F_3^{\mathrm{Z}}\} = x\sum_i \{0, 2e_i g_{\mathrm{A}}, 2g_{\mathrm{V}}g_{\mathrm{A}}\}(q_i - \bar{q}_i), \qquad (9)$$

where e_i is the charge of the quark species i and g_{V} and g_{A} are the weak vector and axial-vector couplings of the quark to the Z boson.

We use the QCDNUM package to compute the structure functions at NNLO which involves convolutions of parton densities with various coefficient functions [36–41]. Note in this respect that the coefficients in Eq. (9) are the same for all up-type (down-type) quarks with $e_i = \frac{2}{3}(\frac{1}{3})$. Thus we can compute NNLO structure functions separately for the sum of up-type and down-type quarks and multiply these by the coefficients afterward. This allows for an efficient calculation of the cross section based on the NNLO computation of only six structure functions.

B. Forward model

In the forward modeling approach, we compute the expected number of events ν in a bin $(\Delta x, \Delta Q^2)$ as an integral of the differential cross section over the full kinematic phase space $\mathcal{D}$. Introducing the short-hand notation $\boldsymbol{u} = \{x, Q^2\}$ this integral can be written as

$$\nu = \mathcal{L}\int_{\Delta\boldsymbol{u}'}\left[\int_{\mathcal{D}} A(\boldsymbol{u}'|\boldsymbol{u})\frac{\mathrm{d}^2\sigma(\boldsymbol{u})}{\mathrm{d}\boldsymbol{u}}\mathrm{d}\boldsymbol{u}\right]\mathrm{d}\boldsymbol{u}', \qquad (10)$$

where the primed (unprimed) variable refers to the reconstructed (true) kinematics. Here $\mathcal{L}$ is the integrated luminosity of the dataset being analyzed, and A is a transformation kernel that takes into account the detector effects and radiative corrections to the QED Born-level differential cross sections, computed as described in Sec. III A. The forward model described here can be implemented for other experiments or independently of PartonDensity.jl as long as the transfer matrix A is provided by the experimental groups as part of their data release.

We assume here that this information is cast in the form of a matrix A that provides a mapping from the Born-level cross section integrated over bins in the kinematic domain to events accumulated in the experimental bins. Denoting by ν_i the counts in an experimental bin i and by λ_j the integrated cross section in a kinematic bin j, Eq. (10) is written as

$$\nu_i = \mathcal{L}\sum_j A_{ij}\lambda_j. \tag{11}$$

Experimental systematic uncertainties are encoded in a set of deviation matrices. The total systematic deviation is then given by a linear combination of these deviations with weights that are included in the set of model parameters θ. In Sec. VI A, we will describe in more detail how the forward modeling is implemented in our present analysis framework.

Because we are analyzing event counts, it is trivial to write down the likelihood as a product of Poisson distributions:

$$\mathcal{L}_D(\theta) = \prod_{\text{bins}} \frac{\nu^n e^{-\nu}}{n!}, \tag{12}$$

where n is the number of events observed in a bin and ν is that predicted by the forward model, using a particular set of model parameter values θ.

Analyzing binned event counts has the advantage that the data points are uncorrelated and have known probability distributions and that there are no problems with including sparsely populated or empty bins. To our knowledge, these features are currently unique to our approach.

IV. TECHNICAL DEVELOPMENTS

We now describe the main numerical and programming steps carried out while developing our analysis code.

A. The SPLINT package

Integrating the cross sections with standard numerical methods like two-dimensional Gauss quadrature turns out to be quite CPU-time-consuming because a sizable sample of NNLO structure functions has to be computed for each integration. To solve this problem, we added to the QCDNUM distribution the SPLINT package to construct cubic interpolation splines of structure functions and cross sections. Sampling from splines is much faster than *ab initio* computation, while SPLINT provides fast routines for cubic spline integration.

A cubic interpolation spline $S(u)$ in the variable u is a piecewise cubic polynomial defined on a strictly ascending set of node points $\{u_i\}$. The four polynomial coefficients in each node bin are adjusted such that the spline coincides at each node u_i with an input value f_i, and is continuous in the first and second derivative; the third derivative is allowed to be discontinuous at the node points. These conditions lead to a set of linear equations in the coefficients, which can be solved if boundary conditions are given on the slopes at the two end points of the spline. We use a spline algorithm that fixes the third derivatives to values estimated from divided differences. A one-dimensional spline $S(u)$ becomes a two-dimensional spline $S(u, v)$ by parametrizing the u coefficients as cubic splines in v.

The SPLINT package constructs splines on a selected set of QCDNUM $x - Q^2$ evolution grid points. Alignment of the nodes to the evolution grid avoids QCDNUM interpolation of a user-given input function $f(x, Q^2)$ when constructing the spline. A coarser node grid gives a faster spline construction but also a less precise spline approximation so that some tuning is necessary to balance speed versus accuracy; see Sec. IV B.

Spline interpolation becomes interesting when many samples are needed of functions that are expensive to compute, such as structure functions and cross sections. For this purpose SPLINT has a special routine for structure function input that makes use of the very fast list-processing capabilities of QCDNUM. Apart from creating splines, the SPLINT package also has routines to integrate these over rectangular bins taking, if necessary, into account the kinematic limit $Q^2 \leq xs$, where $\sqrt{s}$ is the center-of-mass energy of the $e^{\pm}p$ collisions.

Inside SPLINT, the splines are functions of the internal QCDNUM variables $u = -\ln x$ and $t = \ln Q^2$, which introduces a Jacobian $e^{-u}e^{t}$ in the integrals. As a consequence, spline integration is expressed in terms of the fundamental (γ-function) integrals

$$E^{\pm}(z, n) = \int_0^z w^n e^{\pm w} \mathrm{d}w. \tag{13}$$

Partial integration of Eq. (13) yields simple recursion relations between the $E^{\pm}$ at successive values of n so that they can be computed rapidly for all powers 0–3 needed for the term-by-term integration of a cubic polynomial inside a node bin. When a node bin crosses the kinematic limit, we have to integrate both over rectangles and right-angled triangles in the $u - t$ plane. The triangular domains are handled by SPLINT integration over u and Gauss integration over t.

To validate the procedure, we integrated splines over arbitrary rectangles, with and without crossing the kinematic limit, both with SPLINT and with a two-dimensional Gauss integration routine. In all cases, the relative difference was found to be $< 10^{-9}$ with SPLINT running about a factor of 300 faster than Gauss.

For a more detailed description of the integration algorithm in SPLINT we refer to the write-up, which is available from the QCDNUM website.[3]

B. The tuning of QCDNUM and SPLINT

In the range $x > 10^{-3}$ and $100 < Q^2 < 3 \times 10^4$ GeV^2 we have tuned the QCDNUM evolution grid and the SPLINT

[3] https://www.nikhef.nl/user/h24/qcdnum.

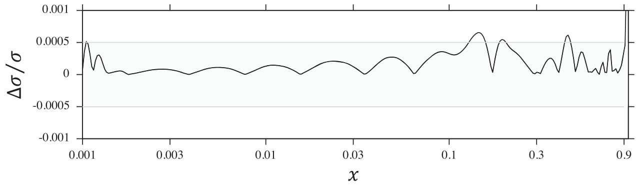


FIG. 1. Estimate of the relative error on the differential cross section versus x at $Q^2 = 100\ \mathrm{GeV}^2$. Above $x = 0.95$ the cross section vanishes, and a relative error becomes ill defined.

spline nodes with the aim to obtain a relative accuracy on the cross sections of better than 5×10^{-4}, with the code still running at an acceptable speed. For this tuning, the cross sections are computed at $\sqrt{s} = 300$ GeV.

In [24] it is recommended to run the QCDNUM evolution on a 100×50 x-Q^2 grid with break points at $x = \{0.2, 0.4, 0.6, 0.8\}$ where the grid density doubles toward larger x at each break point. Comparison with evolution on a 300×150 grid shows a relative PDF accuracy of better than 10^{-4}, except at very large x where the parton distributions vanish.

Because structure functions are slowly varying in x and Q^2, a coarse 22×7 node grid is sufficient to yield interpolation splines with a relative accuracy better than 5×10^{-4}, and the same is true for the cross section spline on a 100×25 node grid; see Fig. 1. The latter grid must be dense since the cross section is strongly varying.

In Table I, we show the CPU time needed when running the tuned code on a 2018 MacBook Pro with an Intel processor. Starting from a set of input PDFs, computing 429 integrated cross sections takes less than 10 ms on such a machine.

C. The QCDNUM.jl wrapper

We have developed the QCDNUM.jl package[4] which is the Julia interface to the QCDNUM fast QCD evolution and convolution routines. The QCDNUM program is written in FORTRAN77, and the interface gives us access to this fast, versatile, and well-tested software in the BAT analysis framework (see Sec. II D).

With QCDNUM.jl, we provide a lightweight but easily usable wrapper, with all the original QCDNUM functionality available to a Julia programmer. Example programs are provided in the documentation, allowing those familiar with QCDNUM to quickly adapt their code. As part of the Julia interface, we also provide a well-documented, high-level interface to implement and visualize the evolution of PDFs that may be appealing to both new and old users of QCDNUM.

TABLE I. CPU cost of computing integrated cross sections.

Subtask	Grid	CPU time [ms]
Evolution	100×50	3.6
Structure function splines (6×)	22×7	2.9
Cross section spline	100×25	2.2
Integration over 429 bins		0.8

The implementation in Julia offers several further advantages. Thanks to `BinaryBuilder.jl`,[5] QCDNUM is automatically compiled and installed in a cross-platform manner, with no actions needed from the user. It is also possible to interface code relying on QCDNUM with Julia 's rich functionality and ever-growing modern package ecosystem. The existing tools for working with Julia code, such as Jupyter notebooks, enable more accessible and reproducible code. As such, the QCDNUM.jl package is a useful addition to the community bringing the use of QCDNUM to a broader audience and modern coding practices.

D. PartonDensity.jl package

We implement a full forward model and interface to BAT in PartonDensity.jl[6] to enable the simulation of data and inference of PDF parameters. Included are the interfaces to QCDNUM and SPLINT, as well as the different PDF parametrizations described in this work. We also implement the forward modeling transfer matrices from the ZEUS experiment, described in Sec. VI A. Within this framework, we provide prior and likelihood definitions that can be used with BAT, as well as tools for visualization. The software is well documented and designed in a modular way to allow extension to other PDF parametrizations and datasets of interest.

Several forms of standard output are available from our code. One useful feature is a text-file summary of the posterior distribution, an example of which is shown in the upper panel of Fig. 2. The marginalized posteriors and global mode values are given for all parameters. Many graphical representations of the output are available in standard form, and others can be easily generated. As an example, we show in the lower panel of Fig. 2 a two-by-two correlation plot that can be made for any subset of the parameters.

V. PARTON DENSITY PARAMETRIZATIONS

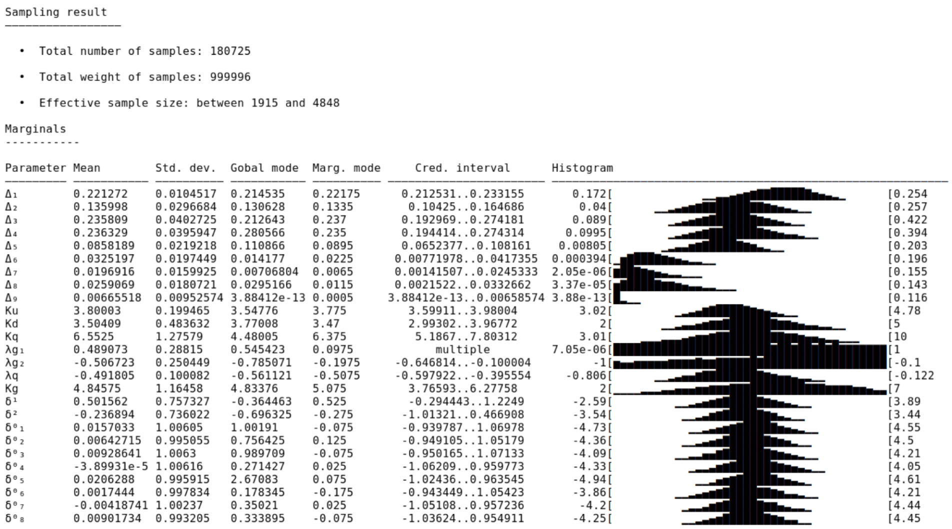
The PDF determination procedure requires a well-defined set of parametrizations for the different parton densities at the input scale Q_0^2 of the evolution. A variety of forms are currently used by the different fitting teams [1–10] which are all of the type

$$xf_i(x) = A_i x^{\lambda_i} F_i(x)(1-x)^{K_i}. \tag{14}$$

[4] https://github.com/cescalara/QCDNUM.jl.

[5] https://binarybuilder.org.

[6] https://github.com/cescalara/PartonDensity.jl.



```
Sampling result
────────────────

  •  Total number of samples: 180725

  •  Total weight of samples: 999996

  •  Effective sample size: between 1915 and 4848

Marginals
-----------

Parameter Mean        Std. dev.  Gobal mode  Marg. mode     Cred. interval       Histogram
───────── ─────────── ────────── ─────────── ──────────  ───────────────────────  ──────────────────────────────────────────────────────
Δ₁        0.221272    0.0104517  0.214535    0.22175       0.212531..0.233155       0.172[                                        [0.254
Δ₂        0.135998    0.0296684  0.130628    0.1335         0.10425..0.164686        0.04[                                        [0.257
Δ₃        0.235809    0.0402725  0.212643    0.237         0.192969..0.274181       0.089[                                        [0.422
Δ₄        0.236329    0.0395947  0.280566    0.235         0.194414..0.274314      0.0995[                                        [0.394
Δ₅        0.0858189   0.0219218  0.110866    0.0895        0.0652377..0.108161    0.00805[                                        [0.203
Δ₆        0.0325197   0.0197449  0.014177    0.0225      0.00771978..0.0417355  0.000394[                                        [0.196
Δ₇        0.0196916   0.0159925  0.00706804  0.0065      0.00141507..0.0245333  2.05e-06[                                        [0.155
Δ₈        0.0259069   0.0180721  0.0295166   0.0115      0.0021522..0.0332662   3.37e-05[                                        [0.143
Δ₉        0.00665518  0.00952574 3.88412e-13 0.0005     3.88412e-13..0.00658574 3.88e-13[                                        [0.116
Ku        3.80003     0.199465   3.54776     3.775        3.59911..3.98004          3.02[                                        [4.78
Kd        3.50409     0.483632   3.77008     3.47         2.99302..3.96772             2[                                        [5
Kq        6.5525      1.27579    4.48005     6.375         5.1867..7.80312          3.01[                                        [10
λg₁       0.489073    0.28815    0.545423    0.0975          multiple           7.05e-06[                                        [1
λg₂       -0.506723   0.250449   -0.785071   -0.1975     -0.646814..-0.100004         -1[                                        [-0.1
λq        -0.491805   0.100082   -0.561121   -0.5075     -0.597922..-0.395554     -0.806[                                        [-0.122
Kg        4.84575     1.16458    4.83376     5.075        3.76593..6.27758             2[                                        [7
δ¹        0.501562    0.757327   -0.364463   0.525         -0.294443..1.2249       -2.59[                                        [3.89
δ²        -0.236894   0.736022   -0.696325   -0.275       -1.01321..0.466908       -3.54[                                        [3.44
δᶿ₁       0.0157033   1.00605    1.00191     -0.075       -0.939787..1.06978       -4.73[                                        [4.55
δᶿ₂       0.00642715  0.995055   0.756425    0.125        -0.949105..1.05179       -4.36[                                        [4.5
δᶿ₃       0.00928641  1.0063     0.989709    -0.075       -0.950165..1.07133       -4.09[                                        [4.21
δᶿ₄       -3.89931e-5 1.00616    0.271427    0.025        -1.06209..0.959773       -4.33[                                        [4.05
δᶿ₅       0.0206288   0.995915   2.67083     0.075        -1.02436..0.963545       -4.94[                                        [4.61
δᶿ₆       0.0017444   0.997834   0.178345    -0.175       -0.943449..1.05423       -3.86[                                        [4.21
δᶿ₇       -0.00418741 1.00237    0.35021     0.025        -1.05108..0.957236        -4.2[                                        [4.44
δᶿ₈       0.00901734  0.993205   0.333895    -0.075       -1.03624..0.954911       -4.25[                                        [4.45
```

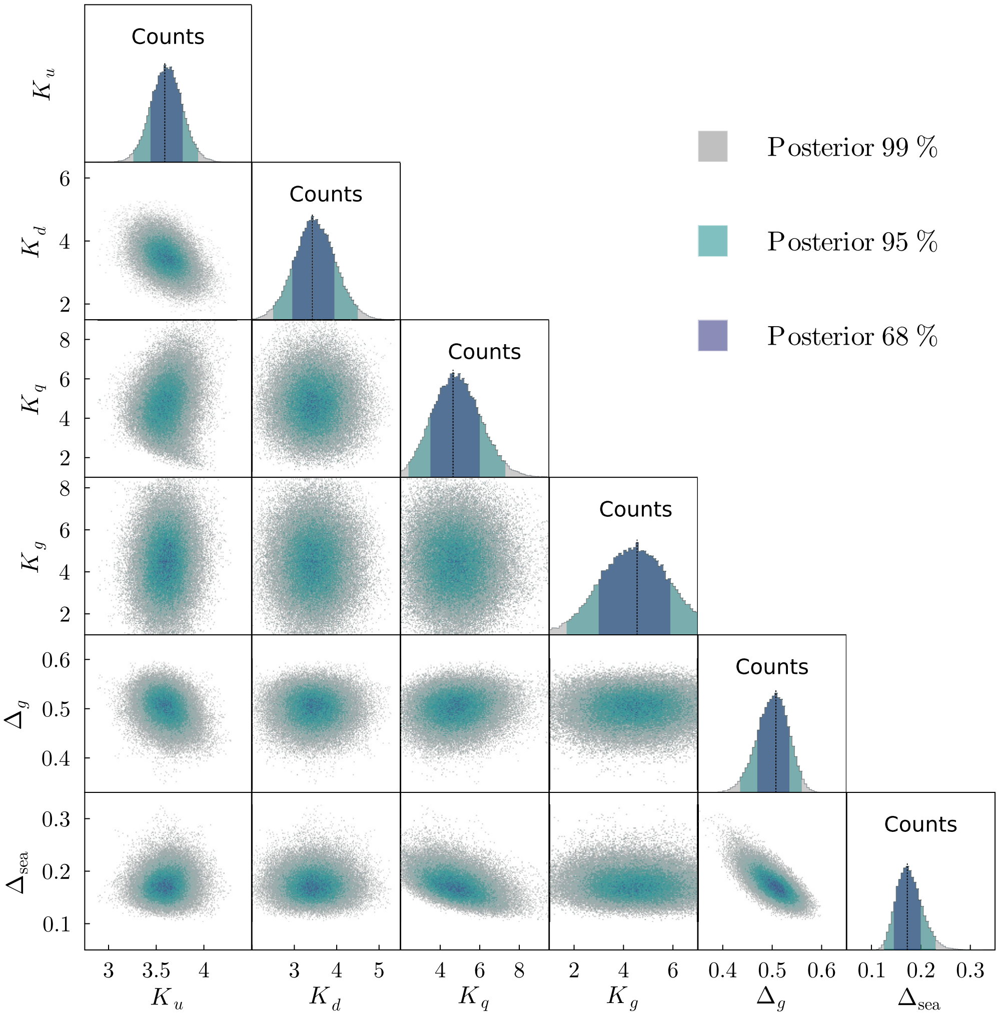


FIG. 2. Upper panel: example of a statistical summary of a 26-dimensional posterior distribution. Lower panel: parameter correlation distributions for a subset of the model parameters. The marginal distributions are shown on the diagonal. Taken from the pseudodata analysis described in Sec. VI.

Each parton species i has its own set of parameters and function F. The behavior of a PDF as $x \to 1$ is largely controlled by its parameter K, while λ controls the behavior as $x \to 0$. The function $F(x)$ interpolates between these two extreme regions, and the parameter A fixes the normalization.

In Sec. III A it is mentioned that the quark and gluon distributions $xf_i(x)$ are required to satisfy the momentum sum rule and quark counting rules as given in Eqs. (5) and (6), respectively. We therefore wish to construct parametrizations that are flexible, positive-definite, easily evaluated, and quickly integrated. We also wish, at this stage in the development of the Bayesian framework, to restrict the number of PDF parameters as much as possible. Given these requirements, we have fully implemented in PartonDensity.jl a beta-distribution parametrization that provides a suitable starting point in a future exploration of a variety of alternatives.

A. Parametrization based on beta distributions

To parametrize the quark densities, it is convenient to write them as valence (q^{v}) and sea (q^{s}) distributions,

$$q + \bar{q} = (q - \bar{q}) + 2\bar{q} = q^{\mathrm{v}} + q^{\mathrm{s}}.$$

We parametrize the valence momentum densities as

$$xq_i^{\mathrm{v}}(x, Q_0^2) = \begin{cases} A_i x^{\lambda_i}(1-x)^{K_i} & \text{for } i = u, d \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

The integral of the number density, ${q^{\mathrm{v}}}_i(x, Q_0^2)$ is finite for $\lambda_i > 0$. Similarly, for the antiquark distributions, we have

$$x\bar{q}_i(x, Q_0^2) = A_i x^{\lambda_{\bar{q}}}(1-x)^{K_{\bar{q}}} \quad \text{for } i = \bar{u}, \bar{d}, \bar{s}, \bar{c}, \bar{b}, \tag{16}$$

where all antiquark flavors share the same x dependence, but have different normalizations A_i. For this parametrization, we require $-1 < \lambda_{\bar{q}} < 0$ so that $x\bar{q}$ is integrable and increasing at low x. Finally, we parametrize the gluon density as the sum of valence and sea contributions:

$$\begin{aligned} xg(x, Q_0^2) &= xg^{\mathrm{v}}(x) + xg^{\mathrm{s}}(x) \\ &= A_g^{\mathrm{v}} x^{\lambda_g^{\mathrm{v}}}(1-x)^{K_g} + A_g^{\mathrm{s}} x^{\lambda_g^{\mathrm{s}}}(1-x)^{K_{\bar{q}}}, \end{aligned} \tag{17}$$

where we set $-1 < \lambda_g^{\mathrm{s}} < \lambda_g^{\mathrm{v}}$ and set further restrictions on these parameters in order to obtain an integrable gluon density with a valence (sea) contribution that decreases (increases) toward low x. Here K_g is an independent parameter, while $K_{\bar{q}}$ is shared with the $x\bar{q}_i$ densities defined in Eq. (16).

For a Beta-distribution $xf(x) = Ax^{\lambda}(1-x)^K$ we can replace the normalization constant by the momentum integral through the relation

$$\Delta = A\frac{\Gamma(\lambda+1)\Gamma(K+1)}{\Gamma(\lambda+K+2)}. \tag{18}$$

The valence sum rule, introduced in Eq. (6), relates the normalizations to the shape parameters by

$$A_i = N_i^{\mathrm{v}}\frac{\Gamma(\lambda_i+K_i+1)}{\Gamma(\lambda_i)\Gamma(K_i+1)} \quad i = u, d, \tag{19}$$

with $N_u^{\mathrm{v}} = 2$ and $N_d^{\mathrm{v}} = 1$.

Using the property $\Gamma(z+1) = z\Gamma(z)$ we find from Eqs. (18) and (19) the following relation between the total momentum Δ carried by the u or d valence quarks and the shape parameters:

$$\Delta_i = N_i^{\mathrm{v}}\frac{\lambda_i}{\lambda_i+K_i+1} \quad i = u, d. \tag{20}$$

B. Prior parameter constraints

Given a set of free PDF parameters, the challenge is to include as many physically motivated constraints on the priors as possible.

For this, it is advantageous to replace the normalization constants, A, which are not straightforward for interpretation, with the momentum fractions, Δ, either by numerically integrating the PDFs or by using Eq. (18) in case the PDFs are parametrized in terms of beta distributions.

In that case Eq. (20) offers two alternatives for the parameters of the up and down valence distributions: (i) leave the shape parameters λ and K free and thereby fix the momentum fraction Δ or (ii) leave Δ free and thereby fix one of the two shape parameters.

The first alternative allows one to specify the shape of the valence densities, but it is not trivial to ensure that $\Delta_d + \Delta_u < 1$ and $\Delta_d < \Delta_u$ over the full support of the prior.

In the analysis of the ZEUS high-x data, we have chosen to fix the low-x shape parameter λ and leave free the high-x shape parameter K and the momentum fraction Δ. In this way we include in the model parameters the complete set of PDF momenta, subject to the sum rule constraint of Eq. (5) given in Sec. III A.

It is convenient to use a Dirichlet distribution as a joint prior for the momenta [18]. A Dirichlet distribution $\mathrm{Dir}(\alpha_1, \ldots, \alpha_k)$ of k independent variables $u_i \in [0, 1]$ lives on a $(k-1)$-dimensional manifold defined by $\sum u_i = 1$ so that the momentum sum rule is automatically satisfied. It is a multivariate generalization of the beta distribution; for instance $\mathrm{Beta}(\alpha_1, \alpha_2)$ of one variable u is the same as $\mathrm{Dir}(\alpha_1, \alpha_2)$ of two variables (u_1, u_2) with $u_1 + u_2 = 1$.

With a suitable choice of the shape parameters α, it is possible to satisfy expectations such as that, asymptotically, gluons and quarks carry approximately the same momentum, that valence up quarks carry about twice the momentum of valence down quarks, and that the heavier quarks carry little momentum.

The spectator counting rules of Brodsky and Farrar [42] give a first expectation for the ranges of the shape parameters $K_u, K_d, K_{\bar{q}}$, and K_g. Furthermore, a body of PDF results available from the literature indicate the preferred range for these parameters.

VI. VALIDATION

To validate our Bayesian tools, we analyzed sets of simulated data and compared the posterior distributions to the parameter input values. Because the tools were initially developed for the analysis [17] of the ZEUS high-x and large-Q^2 data [15], our simulations were restricted to the kinematic range covered by these data. An adequate representation is obtained by using the simple beta parametrization described in the previous sections, at an input scale of $Q_0^2 = 100\ \mathrm{GeV}^2$. Some details of the ZEUS experiment are given below.

A. The ZEUS experiment and systematics

In [15], the ZEUS Collaboration has published e$^{\pm}$p deep-inelastic scattering data covering the range $0.03 \leq x \leq 1$ and $650 \leq Q^2 \leq 20000\ \mathrm{GeV}^2$. These data are unique in providing measurements up to $x = 1$ in the high Q^2 regime where higher-twist effects are absent so that a clean analysis can be performed based on the perturbative QCD evolution equations. In [17], it is shown that the data mainly constrain the parameters of the valence up quark, as will also become apparent in the figures and results given below.

The data are presented as counts in 153 bins, separately for the e^-p and e^+p datasets.

To enable forward modeling, ZEUS has provided in [16] the transfer matrix A as defined by Eq. (11) in Sec. III B. This matrix is written as the product of two matrices, R and T, which account for radiative and reconstruction effects, respectively. Also provided are variations on T to enable the evaluation of systematic uncertainties. The QED-Born level binning of 429 bins is finer than the experimental binning.

The R matrices are 429×429 diagonal and correct the Born-level cross sections for $O(\alpha)$ QED effects.

A reconstruction-matrix element T_{ij} gives the probability that an event generated in x-Q^2 bin j is reconstructed in experimental bin i so that T has a dimension of 153×429.

The uncertainties that need to be accounted for in the ZEUS data are the following.

(1) The uncertainty in the luminosity of each of the e$^+$p and e$^-$p datasets;

(2) The uncertainties related to the imperfect understanding of detector effects such as acceptance, energy resolution, etc. These are completely correlated between the two datasets. Variational matrices T' are provided for eight sources of uncertainty in each of the e$^-$p and e$^+$p datasets.

It turns out that the ZEUS luminosity uncertainty of 1.8% is the most important source of uncertainty in this analysis.

From the matrices provided, we can construct for each source of systematic error the deviation matrix

$$A' = R(T' - T),$$

which corresponds to a 1 standard deviation uncertainty, assuming that the systematic errors are Gaussian distributed. Denoting the 1 standard deviation uncertainty on the luminosity by β_0, and including weight factors $\beta_0, \ldots, \beta_8$ in the set of model parameters θ, Eq. (11) can be written as, in matrix notation,

$$\nu = \mathcal{L}(1 + 0.018 \cdot \beta_0)\left(A + \sum_k \beta_k A'_k\right)\lambda, \qquad (21)$$

where the sum runs over the eight sources of systematic error and the β parameters are taken to be normally distributed with unit width.

B. Simulated data

To validate the analysis framework, we generated pseudodata by computing event predictions ν and event counts n that are Poisson distributed with mean ν, with the input parameters given in Table II. The input parameters are chosen such that the simulated pseudodata are similar to the actual ZEUS data. All β parameters are set to zero so that there are no systematic biases in the simulated data. To study convergence, we have also generated pseudodatasets with the ZEUS luminosity increased by a factor of 50 so that the likelihood tends toward a delta function.

C. Fits to the simulated data

Unless otherwise stated, we use the priors listed in Table III in the fits[7] to the pseudodata. The matrices for describing the transformation from QED Born-level cross sections to the observed level are taken from the ZEUS Collaboration [16] for all tests described in this paper. Tests were performed both with the systematic β parameters fixed to zero, or with them left free. Including the systematic parameters, we have a total of 26 free parameters in our fit. The runtime, in this case, is ~250 k samples/hour on a single core.

[7] In a Bayesian analysis, parameters are not fitted in the usual sense. In the context of this paper, a fitted (or free) parameter is one that has a prior *distribution* assigned to it. Fixed parameters, on the other hand, have single-valued priors, thereby reducing the dimension of the parameter space to be explored.

TABLE II. Parameter values used in the data simulation.

$\boldsymbol{\Delta} \times 10^3$								
u^V	d^V	g^V	g^S	$2\bar{u}$	$2\bar{d}$	$2\bar{s}$	$2\bar{c}$	$2\bar{b}$
228	104	249	249	104	52	10	5	0.5
K_u	K_d	λ_g^{v}	λ_g^{s}	K_g	$\lambda_{\bar{q}}$	$K_{\bar{q}}$	$\boldsymbol{\beta}$	
3.70	3.70	0.50	−0.50	5.0	−0.50	6.0	0	

TABLE III. Priors used in the analysis of the pseudodatasets. There are nine parameters in the vector $\boldsymbol{\Delta}$ and ten in $\boldsymbol{\beta}$. The normal distributions are truncated to the range indicated, and their mean and standard deviation are given in brackets.

	Prior	Range
$\boldsymbol{\Delta}$	Dir(20, 10, 20, 20, 5, 2.5, 1.5, 1.5, 0.5)	[0, 1]
K_u	Normal(3.5, 0.5)	[1, 6.5]
K_d	Normal(3.5, 0.5)	[1, 6.5]
λ_g^{v}	Uniform	[0, 1]
λ_g^{s}	Uniform	$[-1, -0.1]$
K_g	Normal(4, 1.5)	[1, 8.5]
$\lambda_{\bar{q}}$	Uniform	$[-1, -0.1]$
$K_{\bar{q}}$	Normal(4, 1.5)	[1, 9.5]
$\boldsymbol{\beta}$	Normal(0, 1)	$[-5, 5]$

Figure 3 shows two-dimensional prior and posterior distributions for the parameters (Δ_u, K_u) and (Δ_d, K_d) obtained from the nominal and high-luminosity pseudo-datasets. As we have already observed in [17], a comparison of posterior to prior shows a very significant knowledge update for the valence up-quark parameters, and less so for those of the valence down-quark. The knowledge update on the parameters of the gluon and sea distributions (not shown) is also rather limited, as is expected from an analysis of high-x data.

A comparison of the left- and right-hand plots in Fig. 3 shows that the posterior converges to the true values, indicating that there is no appreciable bias introduced by our analysis procedure. We have verified that this is the case for all fitted parameters, although convergence with increasing statistics is slow or absent for those that are ill constrained by the data.

Parameters that are well constrained by the data should be relatively insensitive to the choice of prior. Many different datasets were analyzed using shifted prior choices to determine if these significantly affected the results. Two such tests for the up-valence parameters are shown in Fig. 4. In the left-hand plot, the momentum fraction Δ_u was strongly biased upward borrowing momentum from the gluon while the prior for K_u also had an upward bias. In the right-hand plot, the gluon momentum was strongly biased upward, together with a downward bias of K_u. In both cases, the posterior well reproduced the true values of Δ_u and K_u.

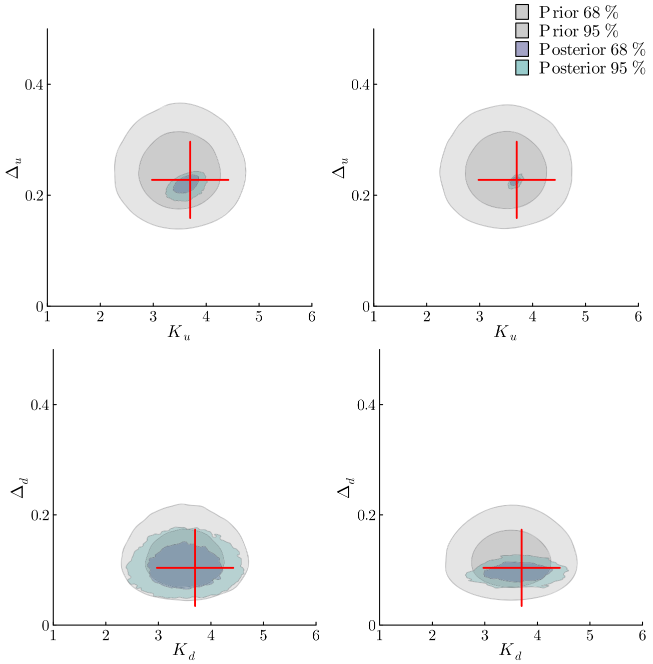


FIG. 3. The prior and posterior probability distributions of (Δ_u, K_u) (upper) and (Δ_d, K_d) (lower) from the nominal (left) and high-luminosity (right) pseudo-data sets. The red crosses indicate the known true parameter values used in generating the pseudo-data.

As a further example of the information made available by the full posterior probability density we plot in Fig. 5 various correlations among the momentum components of the proton obtained from the analysis of the high-luminosity dataset. Here Δ_{sea} is the fractional momentum of the sea (anti)quarks, summed over all quark flavors. Again, it is seen that the true values are nicely reproduced and also that the momentum carried by the valence up quark is very well constrained and weakly sensitive to the other components. The momenta carried by the sea quarks and the gluon density are anticorrelated, as is to be expected since the sea

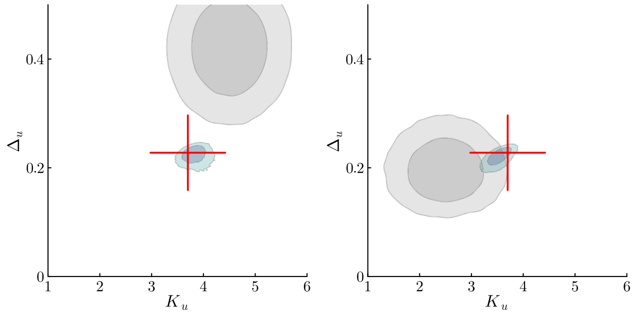


FIG. 4. The posterior probability distributions of (Δ_u, K_u) from two fits to the simulated pseudodata with strongly biased priors (see text). True values are indicated by the red crosses. The legend is the same as that of Fig. 3.

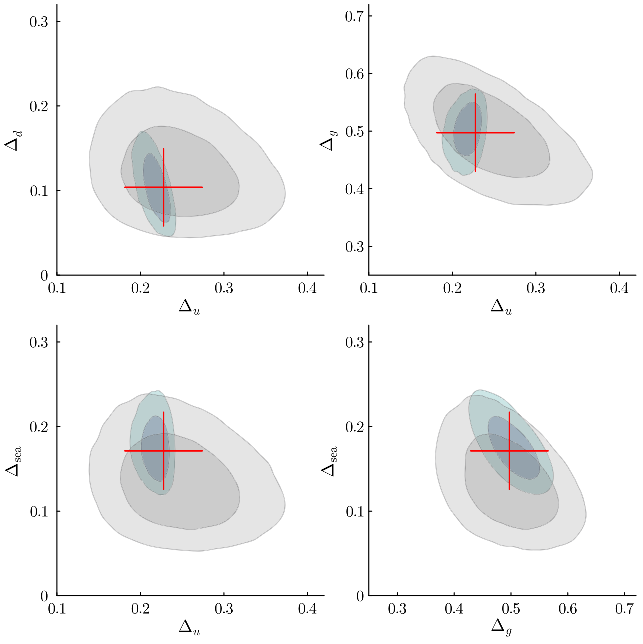


FIG. 5. Correlations between parton momenta from the analysis of the high-luminosity pseudodata. True values are indicated by the red crosses. The legend is the same as that of Fig. 3.

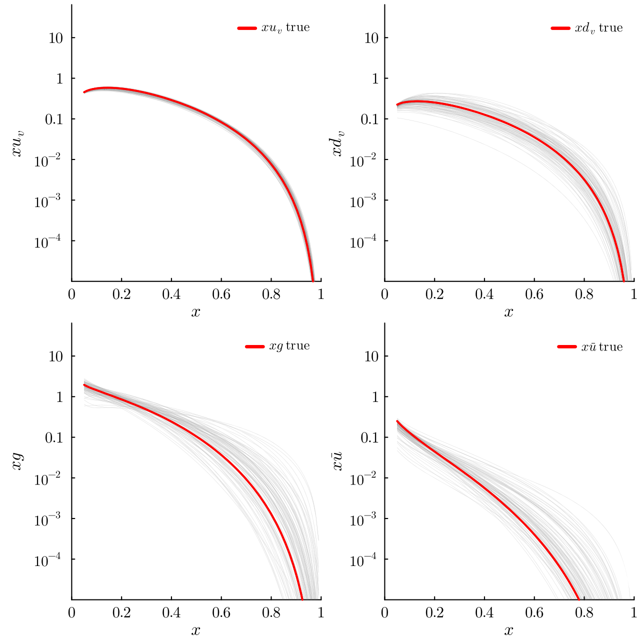


FIG. 6. True parton momentum densities (red curves) compared to those computed from 100 samples of the posterior distribution (gray curves) from the simulated pseudodata at nominal luminosity. The red lines represent the known true densities.

is generated from gluon splitting. For another correlation plot, we refer to Fig. 2 in Sec. IV D.

The results shown up to now were obtained from fits where the systematic parameters were kept fixed to zero. Leaving these parameters free showed that the data hardly constrain them and that they remain uncorrelated, except for the $\mathrm{e}^{+}\mathrm{p}$ and $\mathrm{e}^{-}\mathrm{p}$ luminosity parameters that show a strong correlation, as expected. However, a fit with free systematic parameters is always preferable since that allows for error propagation by marginalizing the posterior over them. However, with the pseudodatasets studied here, this had a minor effect.

We mentioned in Sec. V B an alternative scheme where, instead of fitting all parton momenta, those of the up and down valence quarks are fixed by fitting, instead, their low-x λ shape parameters. In such a fit, we have removed the priors on $\Delta_{u,d}$ and introduced those on $\lambda_{u,d}$ as normal distributions of unit width centered at 0.5 and truncated to a range [0.2, 0.8]. It turned out that the results were very similar to those from the standard parametrization.

In Fig. 6 (note the vertical logarithmic scale) we show input parton distributions used in the generation of the pseudodata compared to those computed from 100 parameter sets randomly sampled from the posterior distribution of the dataset at nominal luminosity. Again, it is seen that the valence up-quark distribution is faithfully reproduced. The valence down-quark distribution is poorly constrained and has a wide spread. The general features of the gluon and up-antiquark distributions are correctly reproduced.

VII. GOODNESS-OF-FIT TESTS

A standard Bayesian analysis does not provide goodness-of-fit tests that compare a single model to data. In fact, a single-model Bayesian analysis yields, as a result, the posterior probability distribution of the model parameters, and no statement is made concerning the validity of the model itself. To investigate model dependence, a choice is to be based on posterior probabilities calculated for several alternative models [43]. Such model selection is beyond the scope of this paper and will be used in a future extension of the framework to investigate the sensitivity to different parametrizations of the input PDFs.

Although not strictly Bayesian, it is possible, and maybe welcome, to provide a goodness-of-fit test based on a single posterior probability distribution. We consider two possibilities below.

A. Posterior predictive check

In a posterior predictive check, the actual data are compared to pseudodata distributions generated from parameter sets that are sampled from the posterior distribution. The quality of the model can then be judged by observing good or bad overlap of the pseudodata and observed data.

A graphical example of such a test is shown in Fig. 7 which shows good agreement. This is not surprising since the data themselves were generated from a given parameter

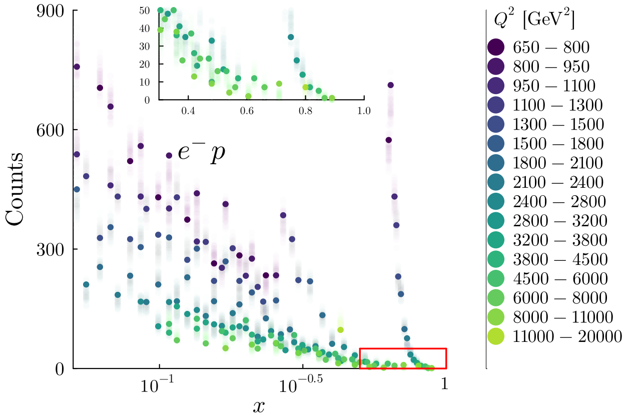


FIG. 7. Event number distributions computed with parameters sampled from the posterior distribution (bands) compared to the simulated e$^-$p scattering pseudodata (dots). For clarity, the box at large x is shown enlarged in the inset.

set, as is described in Sec. VI. Here the figure confirms that the input parameters are well reproduced and that the analysis framework does not introduce significant bias in the result.

B. Posterior mode χ^2 test

A simple goodness-of-fit test is provided by computing the Pearson χ^2, defined by

$$\chi_P^2 = \sum_i \frac{(n_i - \nu_i)^2}{\nu_i},$$

where the sum runs over all bins i with observed event numbers n_i, and expected number of events ν_i as calculated from the posterior global mode parameter values. In the presence of a sizable amount of sparsely populated or empty bins, χ_P^2 does not follow a standard χ^2 distribution. In this case, it is better to calculate χ_P^2 from a large set of simulated pseudodata. In Fig. 8 we show a histogram of the χ_P^2 distribution obtained from many e$^\pm$p pseudodatasets that have ν_i fixed and n_i Poisson distributed around ν. From such histograms, it is straightforward to compute p values by normalizing the histogram and summing the bin contents above the observed value of χ_P^2. In our tests, we have found a fairly flat distribution of p values, as expected.

We have seen in our studies that the maximum posterior probability results do not coincide with a minimum value of χ_P^2. This is not surprising since a maximum posterior is not necessarily a maximum likelihood. Nevertheless, the distribution in Fig. 8 closely resembles that of a standard χ^2 and, as explained in [43,44], small p values can be understood from a Bayesian perspective as implying that the model is open to improvement.

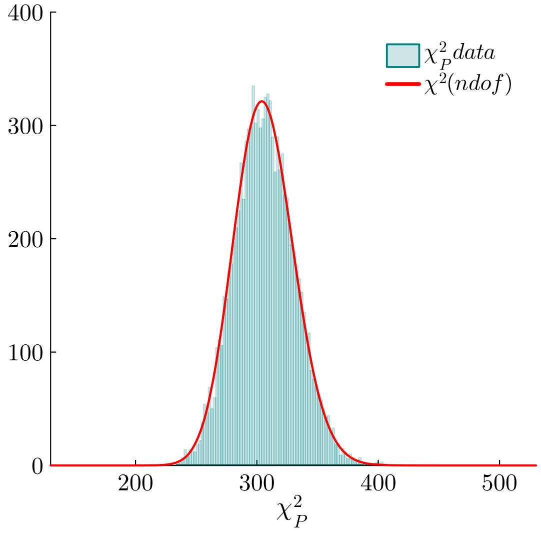


FIG. 8. The χ^2 distribution computed from the combined e$^+$p and e$^-$p analysis results. The curve shows the prediction for 306 degrees of freedom.

VIII. SUMMARY AND OUTLOOK

We have developed a novel parton density analysis code that allows for a full Bayesian posterior probability determination and also supports a forward modeling approach. The open-source code has been thoroughly tested and is now available for distribution. In this paper, we report on its structure, the technical developments that have been made in realizing the code, and a series of validation tests that have been performed. We believe that the code is reliable, well-documented, and easy to use.

To date, the code has been used exclusively for the analysis of high-x and high-Q^2 $e^\pm p$ deep inelastic scattering data. We look forward to also extending the analysis to other datasets, including those reported as differential cross sections at the QED Born level, although such an analysis cannot benefit from the statistical rigor offered by the forward modeling approach. In fact, the information needed to enable such an approach is often not made available in proper form by the experiments.

An important step in achieving this is to make the analysis code run much faster. Here there is ample room for improvement by parallelizing computations through threading or forking, by improving the MCMC sampling efficiency, and by speeding up the QCD evolution of the PDFs.

We also intend to extend the framework to investigate more flexible PDF parametrizations using Bayesian model selection techniques.

ACKNOWLEDGMENTS

The authors would like to thank Richard Hildebrandt for extensive investigations of alternative PDF parametrizations, which we intend to report in a future publication. R. A. acknowledges the support of Savitribai Phule Pune University. F. C. was employed by the Excellence Cluster ORIGINS, which is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2094-390783311, during much of the project duration. The authors would like to thank one of the coauthors, the late Dr. Michiel Botje, who has contributed greatly to this work and will be deeply missed.

[1] T. Cridge, L. A. Harland-Lang, A. D. Martin, and R. S. Thorne, Eur. Phys. J. C **82**, 90 (2022).
[2] S. Bailey, T. Cridge, L. A. Harland-Lang, A. D. Martin, and R. S. Thorne, Eur. Phys. J. C **81**, 341 (2021).
[3] T.-J. Hou *et al.*, Phys. Rev. D **103**, 014013 (2021).
[4] S. Dulat, T.-J. Hou, J. Gao, M. Guzzi, J. Huston, P. Nadolsky, J. Pumplin, C. Schmidt, D. Stump, and C.-P. Yuan, Phys. Rev. D **93**, 033006 (2016).
[5] S. Alekhin, J. Blümlein, and S. Moch, Eur. Phys. J. C **78**, 477 (2018).
[6] S. Alekhin, J. Blumlein, and S. Moch, Phys. Rev. D **89**, 054028 (2014).
[7] R. D. Ball *et al.* (NNPDF Collaboration), Eur. Phys. J. C **82**, 428 (2022).
[8] R. D. Ball *et al.* (NNPDF Collaboration), Eur. Phys. J. C **77**, 663 (2017).
[9] I. Abt *et al.* (H1, ZEUS Collaborations), Eur. Phys. J. C **82**, 243 (2022).
[10] H. Abramowicz *et al.* (H1, ZEUS Collaborations), Eur. Phys. J. C **75**, 580 (2015).
[11] C. Cocuzza, W. Melnitchouk, A. Metz, and N. Sato (Jefferson Lab Angular Momentum (JAM) Collaboration), Phys. Rev. D **104**, 074031 (2021).
[12] N. T. Hunt-Smith, A. Accardi, W. Melnitchouk, N. Sato, A. W. Thomas, and M. J. White, Phys. Rev. D **106**, 036003 (2022).
[13] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, SIAM Rev. **59**, 65 (2017).
[14] O. Schulz *et al.*, SN Comput. Sci. **2**, 210 (2021).
[15] H. Abramowicz *et al.* (ZEUS Collaboration), Phys. Rev. D **89**, 072007 (2014); **106**, 079902(E) (2022).
[16] I. Abt *et al.* (ZEUS Collaboration), Phys. Rev. D **101**, 112009 (2020); **106**, 079901(E) (2022).
[17] R. Aggarwal, M. Botje, A. Caldwell, F. Capel, and O. Schulz, Phys. Rev. Lett. **130**, 141901 (2023).
[18] M. Betancourt, AIP Conf. Proc. **1443**, 157 (2012).
[19] V. N. Gribov and L. N. Lipatov, Sov. J. Nucl. Phys. **15**, 438 (1972).
[20] V. N. Gribov and L. N. Lipatov, Sov. J. Nucl. Phys. **15**, 675 (1972).
[21] L. N. Lipatov, Sov. J. Nucl. Phys. **20**, 94 (1975).
[22] Y. L. Dokshitzer, Sov. Phys. JETP **46**, 641 (1977).
[23] G. Altarelli and G. Parisi, Nucl. Phys. **B126**, 298 (1977).
[24] M. Botje, Comput. Phys. Commun. **182**, 490 (2011).
[25] O. V. Tarasov, A. A. Vladimirov, and A. Y. Zharkov, Phys. Lett. **93B**, 429 (1980).
[26] G. Curci, W. Furmanski, and R. Petronzio, Nucl. Phys. **B175**, 27 (1980).
[27] W. Furmanski and R. Petronzio, Phys. Lett. **97B**, 437 (1980).
[28] W. Furmanski and R. Petronzio, Z. Phys. C **11**, 293 (1982).
[29] S. A. Larin and J. A. M. Vermaseren, Phys. Lett. B **303**, 334 (1993).
[30] K. G. Chetyrkin, B. A. Kniehl, and M. Steinhauser, Phys. Rev. Lett. **79**, 2184 (1997).
[31] M. Buza *et al.*, Eur. Phys. J. C **1**, 301 (1998).
[32] S. Moch, J. A. M. Vermaseren, and A. Vogt, Nucl. Phys. **B688**, 101 (2004).
[33] A. Vogt, S. Moch, and J. A. M. Vermaseren, Nucl. Phys. **B691**, 129 (2004).
[34] R. Ball, V. Bertone, Ma. Bonvini, S. Forte, P. G. Merrild, J. Rojo, and L. Rottoli, Phys. Lett. B **754**, 49 (2016).
[35] R. L. Workman *et al.* (Particle Data Group), Prog. Theor. Exp. Phys. **2022**, 083C01 (2022).
[36] J. Sanchez Guillen, J. L. Miramontes, M. Miramontes, G. Parente, and O. A. Sampayo, Nucl. Phys. **B353**, 337 (1991).
[37] W. L. van Neerven and E. B. Zijlstra, Phys. Lett. B **272**, 127 (1991).
[38] E. B. Zijlstra and W. L. van Neerven, Phys. Lett. B **273**, 476 (1991).
[39] E. B. Zijlstra and W. L. van Neerven, Phys. Lett. B **297**, 377 (1992).
[40] W. L. van Neerven and A. Vogt, Nucl. Phys. **B568**, 263 (2000).
[41] W. L. van Neerven and A. Vogt, Nucl. Phys. **B588**, 345 (2000).
[42] S. J. Brodsky and G. R. Farrar, Phys. Rev. Lett. **31**, 1153 (1973).
[43] F. Beaujean, Phys. Rev. D **83**, 012004 (2011).
[44] A. Caldwell, Ann. Phys. (Berlin) **531**, 1700457 (2019).