



Unbiased determination of DVCS Compton form factors

H. Moutarde^{1,a}, P. Sznajder^{2,b}, J. Wagner^{2,c}

¹ IRFU, CEA, Université Paris-Saclay, 91191 Gif-sur-Yvette, France

² National Centre for Nuclear Research (NCBJ), Pasteura 7, 02-093 Warsaw, Poland

Received: 7 May 2019 / Accepted: 8 July 2019 / Published online: 20 July 2019
© The Author(s) 2019

Abstract The extraction of Compton form factors (CFFs) in a global analysis of almost all deeply virtual Compton scattering (DVCS) proton data is presented. The extracted quantities are DVCS sub-amplitudes and the most basic observables which are unambiguously accessible from this process. The parameterizations of CFFs are constructed utilizing the artificial neural network technique allowing for an important reduction of model dependency. The analysis consists of such elements as feasibility studies, training of neural networks with the genetic algorithm and a careful regularization to avoid over-fitting. The propagation of experimental uncertainties to extracted quantities is done with the replica method. The resulting parameterizations of CFFs are used to determine the subtraction constant through dispersion relations. The analysis is done within the PARTONS framework.

1 Introduction

An intense experimental activity has been dedicated over the last twenty years to the measurements of observables towards a 3D description of the nucleon. Most of the knowledge about the 3D quark and gluon structure is embodied in Generalized Parton Distributions (GPDs) and Transverse Momentum Dependent parton distribution functions (TMDs). The continuing efforts in understanding and determining GPDs and TMDs are partly driven by the quest for nucleon tomography in mixed position-momentum space or pure momentum space (see e.g. Refs. [1,2] and refs. therein), the access to the nucleon energy-momentum tensor [3,4] and the description of the mechanical properties (radial and tangential pressures, energy density, surface tension, *etc.*) of the nucleon [5–8].

Relations between measurements and GPDs or TMDs are derived through factorization theorems, established at

all order in QCD perturbation theory (pQCD). In particular, this connection can be brought under good theoretical control with a careful and systematic check of various pQCD assumptions. This task requires a sophisticated and modular computing machinery with fitting features. In the context of GPDs, the open-source PARTONS framework [9] has been designed and publicly released to fulfill the needs of the experimental and theoretical GPD communities.

From a theoretical point of view, Deeply Virtual Compton Scattering (DVCS) is the cleanest channel to access GPDs. Recent status of the phenomenology of this channel can be found in Ref. [10] and of the related collected data sets in Ref. [11]. The DVCS cross section is described in terms of Compton form factors (CFFs), which are convolutions of GPDs with coefficient functions computed order by order in pQCD. While GPDs are in principle renormalization and factorization scale-dependent, CFFs do not depend on any arbitrary scales, and are the off-forward analogs of the classical structure functions of Deeply Inelastic Scattering (DIS) – with GPDs being the off-forward extensions of the Parton Distribution Functions (PDFs) extracted from DIS.

Global GPD fits over the whole DVCS kinematic domain, from the glue to the valence region, have not been achieved yet. However, a great deal is already known about CFFs, in particular in the valence region. This is indeed precious knowledge: due to the scale dependence, GPDs can only be promoted to the status of quasi-observables thanks to universality (the same GPDs should allow the simultaneous analysis of several independent exclusive processes), while CFFs are genuine observables. Hence measuring CFFs is an interesting task per se. Moreover, having a unified description of all existing DVCS measurements in terms of CFFs is important in view of the definition of future experimental programmes: one has to select an observable to be measured, specify the kinematic region, and evaluate the required uncertainty in order to make substantial progress in the understanding of the DVCS process. This is partic-

^a e-mail: herve.moutarde@cea.fr

^b e-mail: pawel.sznajder@ncbj.gov.pl

^c e-mail: jakub.wagner@ncbj.gov.pl

ularly relevant for future experimental campaigns at Jefferson Lab or for Electron-Ion Collider (EIC) design studies.

As mentioned in Ref. [10], the attempts to determine CFFs from experimental data mostly fall into two categories:

Local fits CFFs are independently determined from measurements between different kinematic bins. This amounts to a sampling of CFFs over the probed kinematic domain. The model-dependence of the result is low, but most of the time, the problem is ill-posed by lack of uniqueness.

Global fits The free coefficients of a CFF parameterization are matched to experimental data. Kinematic bins are no more treated independently. Interpolating between measurements of the same observable on neighboring kinematic bins is feasible. Extrapolating out of the domain constrained by measurements becomes possible, paving the way for impact studies. However the estimation of the systematic uncertainty associated to the choice of a parameterization is an extremely difficult task, and this is a limit to potential impact studies.

A possible solution to the tension between the advantages and drawbacks of these two approaches may be found in Artificial Neural Networks (ANNs) [12]. While being of common use for PDFs today, their role in GPD-related physics has been assessed (to the best of our knowledge) only in the pioneering work of Kumerički, Müller and Schäfer [13]. ANNs offer the joint promise of a great flexibility and of a common description of data sets in different kinematic regions. This difficulty of this last point should not be underestimated. In our previous fit of CFFs from almost all existing DVCS data [14], our physically-motivated choice of CFF parameterizations restricted the scope of our study to the quark sector. There is a need for flexible parameterizations while escaping at the same time the *curse of dimensionality*, and in this respect, ANNs are quite appealing solutions.

We present here the first global fit of most existing DVCS measurements in terms of CFFs described by neural networks. The paper is organized as follows. In Sect. 2 we briefly introduce the DVCS channel, define CFFs and stress their role in dispersion relations. Section 3 is a reminder of ANN technique and of the genetic algorithm. Our implementation of those techniques and the main elements of our analysis are detailed in Sect. 4. We point out the experimental data being used in this fit in Sect. 5, and outline our results with a focus on the DVCS subtraction constant in Sect. 6. This discussion is of major contemporary interest since it is related to the measurability of the distribution of pressure forces in the proton [7,8,15,16]. At last we summarize in Sect. 7.

2 Theory framework

The golden channel in the GPD extraction programme is the leptonproduction of a single real photon off a nucleon:

$$l(k) + N(p) \rightarrow l(k') + N(p') + \gamma(q'). \quad (1)$$

Here, the symbols between parentheses denote the four-momentum vectors of the lepton, l , the nucleon, N , and of the real photon, γ . The amplitude for this process, \mathcal{T} , is given by a sum of amplitudes for two processes having the same initial and final states: \mathcal{T}^{BH} for the purely electromagnetic Bethe-Heitler process and $\mathcal{T}^{\text{DVCS}}$ for the hadronic DVCS process, such as:

$$\mathcal{T} = \mathcal{T}^{\text{BH}} + \mathcal{T}^{\text{DVCS}}. \quad (2)$$

The Bethe-Heitler part can be expressed with a great precision in terms of the nucleon electromagnetic form factors. The DVCS part is generally parameterized by twelve helicity amplitudes [17], or equivalently twelve CFFs. However, in this analysis we restrict ourselves only to four of them, which can be related to the leading-twist, chiral-even GPDs. Those CFFs are usually denoted by \mathcal{H} , \mathcal{E} , $\tilde{\mathcal{H}}$ and $\tilde{\mathcal{E}}$, and nowadays they are the most extensively studied ones in the context of GPD phenomenology. The exploration of other CFFs, which can be related to either higher-twist or chiral-odd GPDs, suffers from the sparsity of data collected in the kinematic domain where the factorization theorem applies.

The cross section for single photon production is usually expressed in terms of four variables: (i) the usual Bjorken variable, $x_{\text{Bj}} = Q^2 / (2p \cdot (k - k'))$, (ii) the square of the four-momentum transfer to the target nucleon, $t = (p' - p)^2$, (iii) the virtuality of the exchanged photon, $Q^2 = -q^2 = -(k' - k)^2$, and (iv) the azimuthal angle between the leptonic (spanned by incoming and outgoing lepton momenta) and production (spanned by virtual and outgoing photon momenta) planes. In the case of a transversely polarized target one also introduces the angle ϕ_S between the leptonic plane and the nucleon polarization vector. It is also convenient to exchange the usual Bjorken variable for the generalized one:

$$\xi = -\frac{(q + q')^2}{2(p + p')(q + q')}, \quad (3)$$

which in the case of DVCS is approximately equal to the skewness variable:

$$\eta = \frac{(q' - q)(q + q')}{(p + p')(q + q')}. \quad (4)$$

In this paper a set of lengthy formulae relating CFFs with cross sections is omitted for brevity, but those can be easily found e.g. in Ref. [18]. The unpublished analytic expressions of BH and DVCS amplitudes by Guichon and Vanderhaeghen

are used in this analysis, as implemented and publicly available in the PARTONS framework [9]. Let us note at this moment that the leptonproduction of a single real photon is sensitive to the beam charge, but also to the beam and target polarization states. This gives rise to exclusive measurements performed with variety of experimental setups, allowing for a combination of cross sections to probe specific sub-processes (BH, DVCS or the interference between them) or combinations of CFFs. Experimental data used in this analysis will be introduced in Sect. 5.

Our goal here is a global analysis of DVCS data avoiding any model-dependency whatsoever. We separately extract the real and imaginary parts of the four aforementioned CFFs in the three-dimensional phase-space of (ξ, t, Q^2) . During the extraction we do not exploit the fact that the real and imaginary parts of a single CFF can be related together by a fixed- t dispersion relation (see Ref. [19] for a study at all orders in pQCD):

$$\text{Re}\mathcal{G}(\xi) = C_G + \frac{1}{\pi} \int_0^1 d\xi' \text{Im}\mathcal{G}(\xi') \left[\frac{1}{\xi - \xi'} \mp \frac{1}{\xi + \xi'} \right], \tag{5}$$

Here, $\mathcal{G} \in \{\mathcal{H}, \mathcal{E}, \tilde{\mathcal{H}}, \tilde{\mathcal{E}}\}$ denotes a single CFF with t and Q^2 dependencies suppressed for brevity, one has minus sign for $\mathcal{G} \in \{\mathcal{H}, \mathcal{E}\}$ and plus sign for $\mathcal{G} \in \{\tilde{\mathcal{H}}, \tilde{\mathcal{E}}\}$ in the square brackets, and where C_G is a subtraction constant associated to the corresponding CFF (and GPD), where:

$$C_H = -C_E, \tag{6}$$

$$C_{\tilde{H}} = C_{\tilde{E}} = 0. \tag{7}$$

The dispersion relation is only used afterwards to determine the subtraction constant from the extracted CFF parameterizations. This quantity has on its own important interpretation in terms of strong forces acting on partons inside the nucleon as shown in the recent review Ref. [7]. We also use the requirement of having C_G independent on ξ as a consistency check of our analysis.

Although hadron tomography [20–22] is a very important motivation for the phenomenology of the DVCS process, in the present analysis we restrict ourselves to (quasi) model-independent statements. Therefore we will only sketch what we can foresee from the obtained results and leave detailed quantitative tomographic interpretation for future studies.

3 Methodology

To make this paper self-contained, we remind here some elements of the theory and practice of ANNs and genetic algorithms, and introduce the associated terminology.

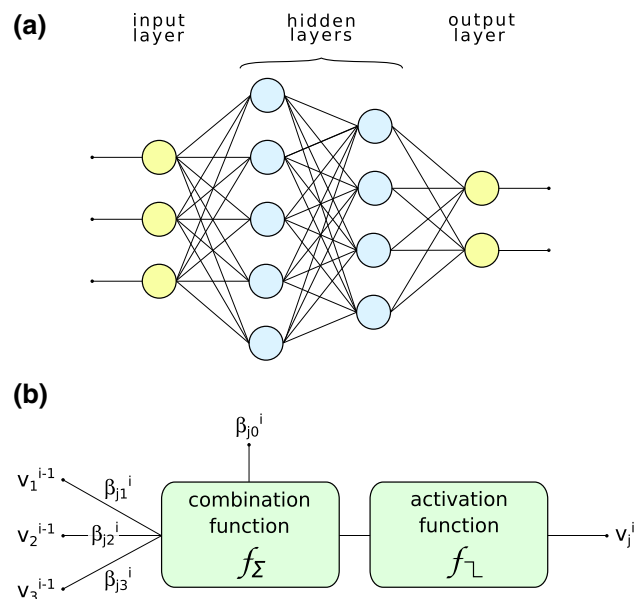


Fig. 1 **a** Example of a typical feed-forward neural network. The network processes three input variables to give two output values ($3 \rightarrow 2$), with the architecture made out of two hidden layers. The first hidden layer contains five neurons, while the second one is made out of four neurons. **b** Schematic illustration of a typical neuron. In this example it is connected to three different neurons belonging to a previous layer of the network. For the explanation of symbols see the text

3.1 Artificial neural networks

ANNs [12] are systems processing information. What distinguishes those systems from other information techniques is a unique inspiration from Nature, namely by biological neural networks making human and animal brains. All neural networks have a similar structure made out of simple, but highly connected elements called neurons. In this work we will exclusively focus on feed-forward neural networks, but many other types of ANNs exist. Their usefulness in the study of hard exclusive reactions has not been assessed yet.

An example of a typical feed-forward neural network structure is shown in Fig. 1a. The data are processed layer-by-layer. One can distinguish three type of layers: (i) one input layer, storing and distributing input information, being the first layer of the network, (ii) a number of hidden layers processing information and (iii) one output layer, aggregating and returning output information, being the last layer of the network. The number of hidden layers is a parameter of the network architecture.

In machine learning neurons share a similar structure shown in Fig. 1b. A single neuron processes information by evaluating two functions: (i) combination function, $f_{\Sigma,j}^i(\vec{\beta}_j^i, \vec{v}^{i-1})$, and (ii) activation function, $f_{L,j}^i(x)$, so that the overall neuron’s output, v_j^i , is given by:

$$v_j^i = f_{\Sigma,j}^i \left(f_{\Sigma,j}^i(\vec{\beta}_j^i, \vec{v}^{i-1}) \right). \quad (8)$$

Here, the index i identifies a specific layer of the neural network, while j denotes a single neuron in that layer. The vector $\vec{\beta}_j^i$ contains a so-called *bias*, β_{j0}^i , and all *weights* associated to the input nodes, β_{jk}^i , with $k = 1, 2, \dots$. The vector \vec{v}^{i-1} contains values returned by neurons in the previous layer, v_k^{i-1} , where again $k = 1, 2, \dots$. The role of the *combination function* is the aggregation of input information to a single value to be processed by the *activation function*. A typical combination function is based on the dot product of two vectors:

$$f_{\Sigma,j}^i(\vec{\beta}_j^i, \vec{v}^{i-1}) = \beta_{j0}^i + \beta_{j1}^i v_1^{i-1} + \beta_{j2}^i v_2^{i-1} + \dots \quad (9)$$

The activation function gives an answer returned by the neuron. It can be a numeric value, but it can be also a binary answer (“yes” or “no”), which is typically used in classification applications. Many types of activation functions exist. In this work we use the hyperbolic tangent function for neurons in the hidden layers,

$$f_{\Sigma,j}^i(x) = \tanh(x), \quad (10)$$

and the linear function (identity) for neurons in the output layer,

$$f_{\Sigma,j}^i(x) = x. \quad (11)$$

We illustrate both functions with Fig. 2. For the hyperbolic tangent function one can spot an active part near $-1 < x < 1$ where the signal is processed almost linearly. Outside this range a saturation appears, so that the input signal is blocked, *i.e.* it is not processed by a neuron in an effective way. Those features put extra constraints on input data to be processed by the network. In particular, those data must be normalized to the range of $(-1, 1)$, so they can be processed by the first layer of the network.

Biases and weights are free parameters of neural networks. Instead of predefining those parameters a training is done, and only during that training biases and weights are fixed to accommodate the network’s answer to the stated problem. Such training is done with a *training sample*, which should be as representative as possible for the problem under consideration. Both input and output information are known for the training sample.

In a wide variety of applications neural networks are trained with the back-propagation algorithm [23]. This algorithm minimizes the so-called *cost function* (e.g. root mean square error, RMSE) against the training sample. Such a minimization usually is straightforward, as typically the cost function is analytically differentiable with respect to each parameter of the network, so that its gradient is known in the space of those parameters. However, there are cases where

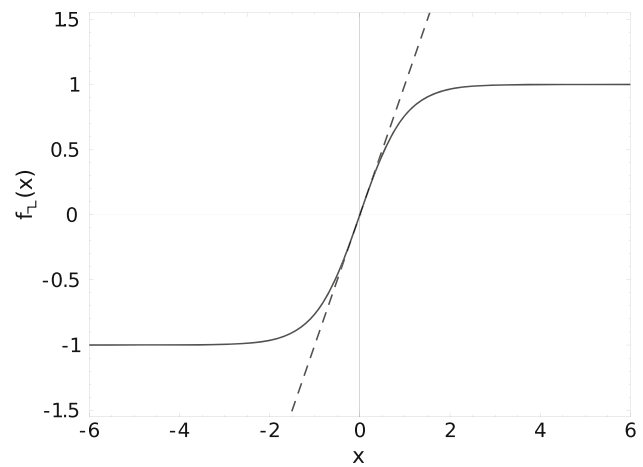


Fig. 2 Hyperbolic tangent (solid curve) and linear (dashed curve) activation functions

the back-propagation algorithm (or similar) cannot be used. Those are in particular problems with the differentiation of the cost function being practically difficult. In those cases one can evaluate the gradient numerically. Eventually, one can use the genetic algorithm [24], which we will describe in the next paragraph. It is a popular replacement for the back-propagation algorithm in the context of machine learning. In particular, the genetic algorithm is used in this analysis, as a differentiation of cross sections and other observables with respect to the free parameters of CFF parameterizations is not straightforward.

3.2 Genetic algorithm

The genetic algorithm [24] is a heuristic technique used for search and optimization purposes. It is applicable in particular whenever standard minimization techniques fail because of the complexity of the problems under consideration. Because of that, but also because of a unique ability to avoid local minima, the genetic algorithm is commonly used in physics, mathematics, engineering and manufacturing. It is frequently used for the learning of neural networks, where it supervises the search of biases and weights against the training sample.

The genetic algorithm mimics mechanisms of reproduction, natural selection and mutation, all being the cornerstones of the evolution process known from Nature. It utilizes nomenclature inspired by the natural evolution process. A *population* is a set of possible solutions to the stated problem. A single solution is referred to as a *candidate* and it is characterized by a set of *genes* making up a *genotype*. In the following, we will thus speak equivalently of candidates and genotypes. The usability of single candidates in solving problems is examined by a *fitness function*, whose argument is the genotype. To help the reader understand this nomen-

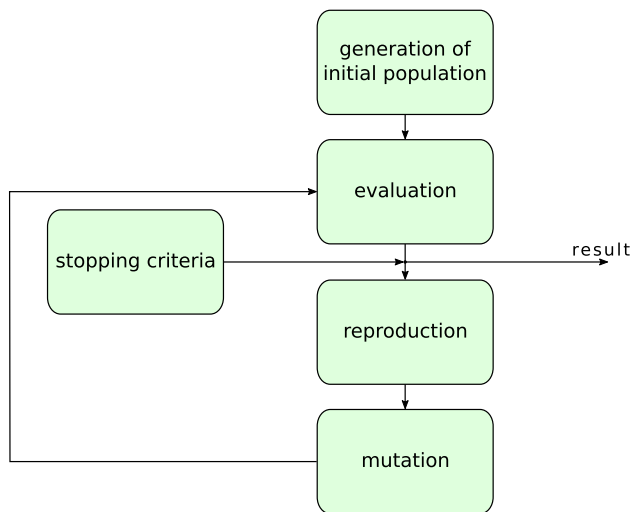


Fig. 3 Scheme of the genetic algorithm

clature we explain it with an example of a multidimensional fit to experimental data. In such a fit a single fitted parameter may be identified as one gene, while a complete set of those parameters as the genotype. The ability to describe experimental data by a given set of parameters is typically examined with a χ^2 function, which in this case we will refer to as the fitness function. A set of possible solutions makes the population. We point out that in general the implementation of introduced terms strongly depends on the problem under consideration.

The genetic algorithm is iterative, that is, it consists of several steps being consecutively executed in a loop, until one of well defined conditions, referred to as *stopping criteria*, is fulfilled. The algorithm is described with the help of Fig. 3. (i) In the first step an initial population is generated. A number of candidates making up this initial population is a parameter of this algorithm. The creation of a new candidate consists for instance of the random generation of its genotype with gene values laying in defined ranges. (ii) Each candidate is evaluated by acting on its genotype with the fitness function. The resulting fitness score indicates how good the candidate is accommodated to the stated problem. Then, all the candidates are sorted according to those scores. The one having the most preferable score is considered as the best one. (iii) Stopping criteria are checked and the algorithm is stopped if one of them is fulfilled. In such a case the genotype of the best candidate is considered as the solution. A possible stopping criterion is for instance a threshold on the fitness score of the best candidate or a threshold on the number of iteration epochs that has passed already. (iv) A fraction of candidates being identified by the worst fitness scores is removed from the population and forgotten. Their place in the population is taken by new candidates carrying genes of the remaining candidates. That is, two or more remaining candidates pass

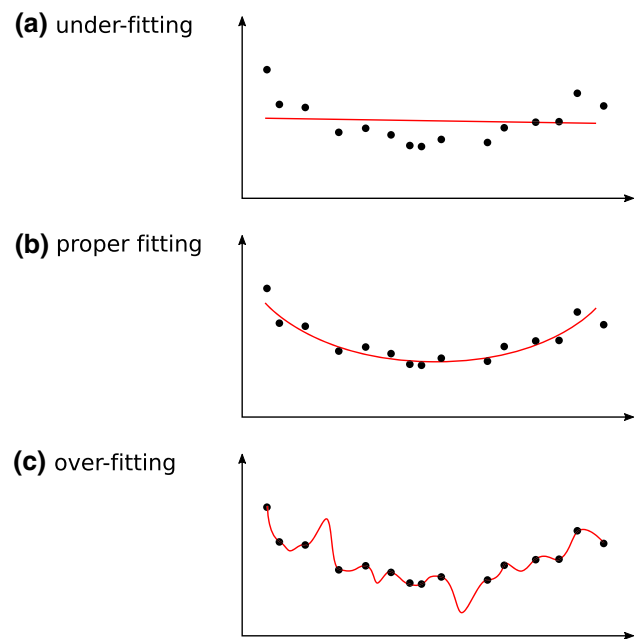


Fig. 4 Illustration of under-fitting, proper fitting and over-fitting. Training data are represented by points, while the effect of a training procedure is represented by red solid curves. The same training data are used in all three cases

a fraction of their genotypes to a new candidate in a hope of creating even a better adapted individual. This step is known as the reproduction or cross-over. (v) A random fraction of genes in the population is modified. This modification may consist of the random generation of a new gene value within a specified range (as during the initialization) or it can be a small shift with respect to the original gene value. The purpose of this step, which is known as the mutation, is to keep the population diverse. The mutation probability cannot be too high, as in such a case the genetic algorithm becomes a random search. The mutation ends a single epoch, so that the algorithm proceeds with the evaluation as the next step.

3.3 Regularization

Two extreme cases of failure in the learning procedure are known as *under-fitting* and *over-fitting*. A compact explanation of those effects goes as follows. Under-fitting results in a lack of success in describing the training sample, as well as any other sample, while over-fitting is having a model being too flexible, which is not able to describe data other than the training sample. Over-fitting usually happens either because the model memorizes a noise present in the training sample, or because it becomes too flexible in a range between training points. The extrapolation does not count here, as neural networks typically become pliant outside ranges covered by the training sample. Both effects, together with the case of proper training, are graphically illustrated in Fig. 4. At this

point we should stress that both under-fitting and over-fitting are not specific features of neural network training, but are known and common problems met in data fitting whenever the number of free parameters becomes large.

Regularization is the introduction of additional information to improve the resolution of badly behaved problems. In the case of neural networks, the regularization helps to avoid over-fitting. In the following we describe the *early stopping* regularization that is used in this analysis, however many other types of regularization exist. We note that in this analysis we have found regularization techniques using a penalty term in the cost function impractical. Such techniques, like Lasso or Ridge regressions [23], make the values of neural network weights small and as a consequence replicas similar to each other (the replication method will be introduced in Sect. 3.4). While this may be seen as a feature in some analyses, in this one it does not lead to a reliable estimation of uncertainties, as replicas are not spread significantly in a domain that is not sufficiently constrained by data.

Our procedure of dealing with over-fitting makes the training *supervised*. It means that we randomly divide our initial sample into two independent sub-samples, which in the following we will refer to as *training sample* and *validation sample*. While the purpose of the training sample remains the same, *i.e.* it is used for the training, the validation sample is exclusively used for the detection of over-fitting.

As a principle, too short a learning leads to under-fitting, while a too long one is responsible for over-fitting, unless the network’s architecture prevents it. Therefore, between those two extremes there must be an optimal epoch in the training corresponding to a most favorable solution. The search of such an optimal epoch is the goal of the early stopping technique. The principle of this method is to monitor the validation sample with the cost function, f_{cost} , and to stop the training whenever the value of this function starts to grow. This idea is illustrated in Fig. 5. In practice, such a growth is difficult to be detected due to fluctuations of the cost function values. Therefore, various quantities are used for its detection. In this analysis we monitor the following quantity evaluated for the test sample, which is inspired by Ref. [25]:

$$f_{\text{stop}}(t_i) = \frac{f_{\text{cost}}^{\text{av}}(t_i)}{\min_{j < i} f_{\text{cost}}^{\text{av}}(t_j)} - 1. \tag{12}$$

Here, t_i and t_j denote different epochs in the training. In the numerator of Eq. (12) one has the value of the cost function for the epoch under consideration, and in the denominator the minimal value of this function obtained in all previous epochs. To minimize the impact of random fluctuations on $f_{\text{stop}}(t_i)$ the cost function is averaged using the running average technique:

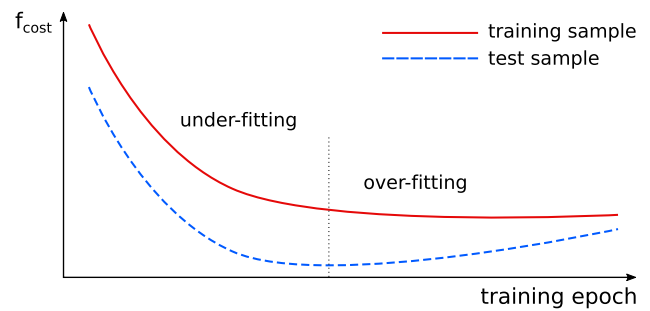


Fig. 5 Demonstration of early stopping regularization technique: the cost function for training (red solid curve) and test (blue dashed curve) samples as a function of training epoch

$$f_{\text{cost}}^{\text{av}}(t_i) = \frac{1}{2n + 1} \sum_{j=-n}^n f_{\text{cost}}(t_i - j), \tag{13}$$

where n is the length of a strip of numbers that is used for the averaging. In this analysis $n = 25$. With those definitions $f_{\text{stop}}(t_i) > 0$ indicates a need of stopping the training. However, as fluctuations of the cost function cannot be completely avoided we stop the training only when $f_{\text{stop}}(t_i) > 0$ for more than one hundred consecutive iterations. We take the set of parameters obtained in the first iteration of such a sequence as the result of our training procedure. In rare cases the stopping criterion is not fulfilled during the whole training procedure. In those cases the result is obtained from the last allowed epoch of this procedure, $t_{\text{max}} = 1000$.

3.4 Replica method

As in our last analysis [14], the *replica method* is used to propagate uncertainties coming from experimental data to CFF parameterizations. In addition to our nominal extraction of CFFs, the fit is repeated one hundred times, each time independently modifying the central values of experimental points using the following prescription:

$$v_{ij} \pm \Delta_{ij}^{\text{tot}} \xrightarrow{\text{replica } k} \left(\text{rnd}_k(v_{ij}, \Delta_{ij}^{\text{tot}}) \pm \Delta_{ij}^{\text{tot}} \right) \times \text{rnd}_k(1, \Delta_i^{\text{norm}}). \tag{14}$$

Here, v_{ij} is the measured value associated to the experimental point j coming from the data set i . It is linked to statistical, $\Delta_{ij}^{\text{stat}}$, systematic, Δ_{ij}^{sys} , and normalization, Δ_i^{norm} , uncertainties. The latter one is correlated across bins of data, like for instance uncertainties related to beam and target polarization measurements. The total uncertainty, which is used to evaluate χ^2 function utilized in this analysis to perform the fit, is given by:

$$\Delta_{ij}^{\text{tot}} = \sqrt{\left(\Delta_{ij}^{\text{stat}}\right)^2 + \left(\Delta_{ij}^{\text{sys}}\right)^2}. \tag{15}$$

The generator of random numbers following a specified normal distribution, $f(x|\mu, \sigma)$, is denoted by $\text{rnd}_k(\mu, \sigma)$, where k both identifies the replica and is a unique random seed.

4 Implementation

4.1 Neural network architecture

In this subsection we describe the architecture of the used neural networks. In particular a determination of the number of hidden neurons in those networks is described. We start with a general statement: any neural network is a container to store information and as such it cannot be too small to serve that purpose. It can be larger than needed though, however such neural networks require a careful regularization to avoid over-fitting and their training is more time-consuming. We note that usually the optimal network’s architecture is selected prior the training, however there exist regularization techniques like *dropout* [26], which alter an initial architecture during the training to obtain best results. In those techniques a number of neurons is either added or dropped as the training proceeds.

We extract four CFFs, each one being a complex quantity having both real and imaginary parts. Although the real and imaginary parts of the a given CFF can be connected together *via* a dispersion relation (with the subtraction constant being involved), in the extraction presented in this paper we keep them fully independent quantities. The dispersion relation is only used for an interpretation of the obtained results, which is presented in Sect. 6. Therefore eight independent neural networks are used.

The architecture of a single network used in this analysis is shown in Fig. 6. It consists of three input neurons, one hidden layer containing six neurons, and one output neuron. Both input and output variables of the network are linearized and normalized (as much as possible). The linearization is achieved with a logarithmic projection and ξ pre-factors:

$$\xi' = \log_{10} \xi, \tag{16}$$

$$t' = t, \tag{17}$$

$$Q^{2'} = \log_{10} Q^2, \tag{18}$$

$$\text{Re}\mathcal{G}' = \xi \text{Re}\mathcal{G}, \tag{19}$$

$$\text{Im}\mathcal{G}' = \xi \text{Im}\mathcal{G}, \tag{20}$$

where the prime symbol is used to distinguish between original and linearized variables. For the normalization we utilize the min-max method:

$$v'' = -1 + 2 \frac{v' - v'_{\min}}{v'_{\max} - v'_{\min}}, \tag{21}$$

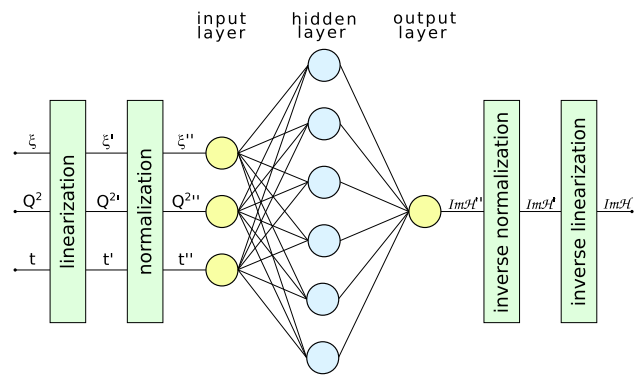


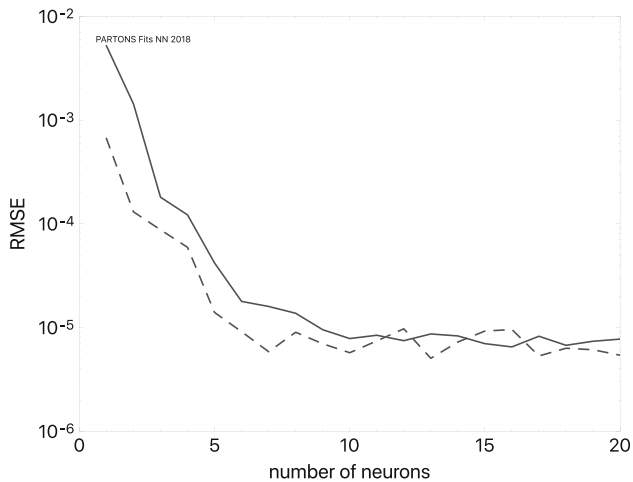
Fig. 6 Scheme of a single neural network that is used in this analysis to represent either the real or the imaginary part of a single CFF

where v' and v'' is a given linearized variable before and after the normalization, respectively, and where v'_{\min} and v'_{\max} are normalization parameters specified in Table 1. The min-max values used for the normalization of ξ' , t' and $Q^{2'}$ have been selected to cover the target phase-space, while those for $\text{Re}\mathcal{G}'$ and $\text{Im}\mathcal{G}'$ roughly correspond to $v'_{\min, \text{GK}} - \Delta v'_{\text{GK}}$ and $v'_{\max, \text{GK}} + \Delta v'_{\text{GK}}$ values, where $\Delta v'_{\text{GK}} = v'_{\max, \text{GK}} - v'_{\min, \text{GK}}$. Here, $v'_{\min, \text{GK}}$ and $v'_{\max, \text{GK}}$ are min-max values found in a poll of CFFs evaluated from the Goloskokov–Kroll (GK) GPD model [27–29] for the experimental data kinematics that are used in this analysis, see Sect. 5. Both linearization and normalization significantly improve the performance of ANNs. We point out that the ranges specified in Table 1 are not absolute, *i.e.* the networks may still reasonably describe data covering the exterior of domains defined by v'_{\min} and v'_{\max} values. In particular in this analysis the activation function of output neurons is set to the identity and thus does not show any saturation effects.

The number of neurons in the hidden layer is determined with a benchmark sample made out of one thousand CFF points evaluated with the GK GPD model. Those CFF points are randomly generated in a broad range of $10^{-6} < \xi < 1$, $0 \text{ GeV}^2 < |t| < 1 \text{ GeV}^2$ and $1 \text{ GeV}^2 < Q^2 < 100 \text{ GeV}^2$. We have checked how many neurons in the hidden layer are needed to provide a good description of the benchmark sample. This test is performed with the FANN library [30] for the neural network implementation and with the *quickprop* training algorithm [31] available in that library. The post-training RMSE for CFF \mathcal{H} is shown in Fig. 7 as a function of the number of hidden neurons. From this figure one may conclude that the addition of a new neuron to the network made already out of six neurons does not significantly improve the performance. We note that a high number of neurons slows down the training and makes the regularization more difficult. That is why in this analysis the number of hidden neurons in each network is set to six, which we consider to be a sufficient number.

Table 1 Values used in the scaling of input and output variables

v'	v'_{\min}	v'_{\max}	v'	v'_{\min}	v'_{\max}
$\log_{10} \xi$	-6	1	t	-1	0.5
$\log_{10} Q^2$	-1	2			
$\xi \text{Re} \mathcal{H}$	-1	1.5	$\xi \text{Im} \mathcal{H}$	-4	6
$\xi \text{Re} \tilde{\mathcal{H}}$	-0.6	0.9	$\xi \text{Im} \tilde{\mathcal{H}}$	-1	1.5
$\xi \text{Re} \mathcal{E}$	-1.5	1	$\xi \text{Im} \mathcal{E}$	-6	4
$\xi \text{Re} \tilde{\mathcal{E}}$	-120	180	$\xi \text{Im} \tilde{\mathcal{E}}$	-8	12

**Fig. 7** Average root mean square error (RMSE) for the neural network describing the benchmark sample (see the text) as a function of the number of hidden neurons in that network for the real (solid line) and imaginary (dashed line) parts of CFF \mathcal{H} . The RMSE values correspond to the normalized variables, see Table 1

4.2 Initial fit

Without any improvement the fit either does not converge or converges very slowly. This is due to a large number of parameters to be constrained in the training and a widely opened phase-space for CFF values, *cf.* Table 1. To overcome this difficulty the network is initially trained on values obtained in a local extraction of CFFs. That is, CFF values multiplied by ξ are independently extracted in each kinematic bin of data, analogously as in Refs. [15, 32, 33], and then used in a training based on the back-propagation algorithm.

The local extraction procedure includes all four CFFs, so effectively eight values per each kinematic bin of (x_{Bj}, t, Q^2) are fitted. This may make a single fit underconstrained, in particular if a given kinematic bin is populated only by a single experimental point. To overcome this difficulty it is imperative to set sensible ranges in which the fitted values are allowed to vary. In this analysis those ranges are set to $(5 \times v'_{\min}, 5 \times v'_{\max})$, where the values of v'_{\min} and v'_{\max} are specified in Table 1. The ranges are wider than those used in the construction of our neural networks, which prevents from

introducing a bias on the final extraction of CFF parameterizations.

The initial fit reduces the χ^2 value per single data point from $\mathcal{O}(10^6)$ to $\mathcal{O}(10)$. The further minimization, which reduces the same quantity to $\mathcal{O}(1)$, is done by our genetic algorithm minimizer, where values of weights and biases obtained in the initial fit are used as starting parameters.

4.3 Genetic algorithm

The parameters of our genetic algorithm minimizer are as follows. The gene values (weights and biases) may vary between -10 and 10 . The population consists of 1000 candidates and the fraction of candidates left after the selection process (survivors) is 30%. We consider two types of mutation and each of them occurs with the same probability of 0.1% per gene. The mutation type-*A* is a random generation of gene values from the range of -10 and 10 . This type of mutation keeps the population diverse. The mutation type-*B* is a small shift of gene values. This shift is randomly generated from a distribution of gene values in the population. This distribution is made for the modified gene. The mutation type-*B* provides a fine-tuning of fitted parameters. The fraction of 20% and 10% best candidates in the population is resistant to the mutation type-*A* and type-*B*, respectively, so the best results are not destroyed by the mutation process.

We demonstrate the performance of our genetic algorithm minimizer with Fig. 8, where values of a single gene are plotted against the training epoch. *Intensity in population* equal 1 means that all candidates in the population share the same gene value. Low values of this quantity indicate that only few or none candidates share the same gene value. One can see few features in this plot being typical for a performance of genetic algorithms: (i) the diversity of values is large at the beginning of minimization, which means that the whole phase-space available is equally scanned to find a cost function minimum, (ii) usually, few such local minima are found at the beginning of minimization and simultaneously explored by the algorithm for a number of iterations, (iii) close to the actual minimum the algorithm concentrates on its neighborhood, trying to perform a fine-tuning of fitted parameters, (iv) even at this stage of minimization, the whole phase-space is homogeneously scanned due to the mutation process to eventually find a new minimum and to avoid the convergence to a local one.

4.4 Regularization

A typical distribution of the cost function as a function of the training epoch is shown in Fig. 9. In addition, we show the distribution of the f_{stop} quantity that we use to detect over-fitting, see Eq. (12). With our stopping criterion:

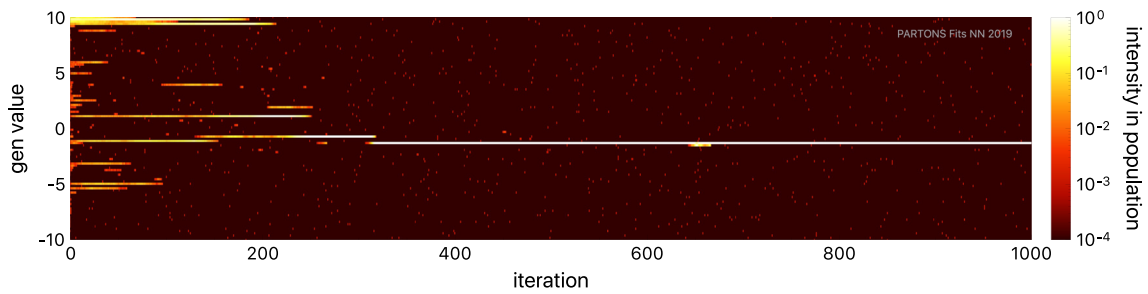


Fig. 8 Diversity of gene values in the whole population against the training epoch for a single gene. For more details see the text

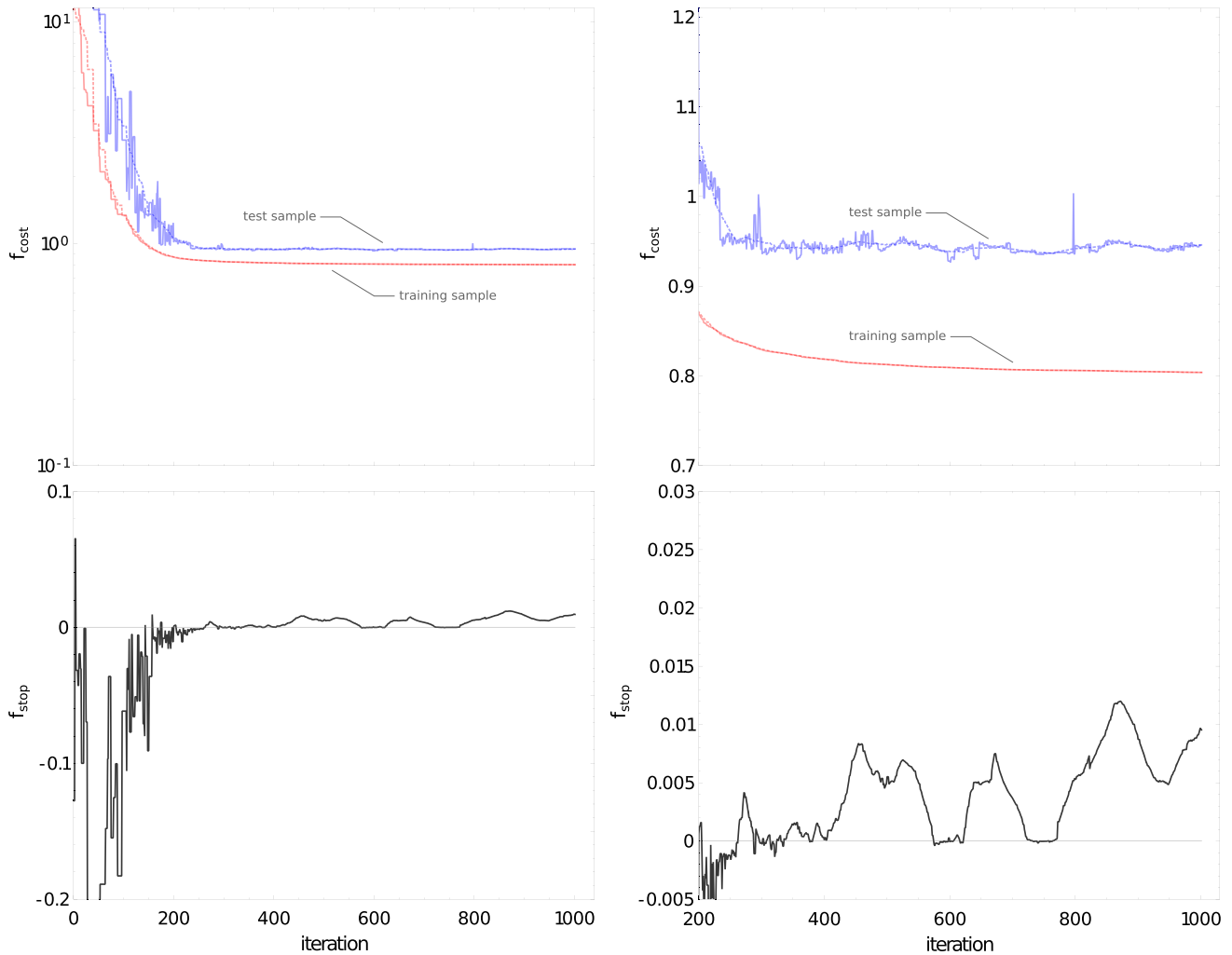


Fig. 9 Cost function (top plots) and f_{stop} quantity (bottom plots) used to detect over-fitting in the regularization method based on the early stopping technique, see Sect. 3.3, as a function of the training epoch. The left plots show dependencies in the full range of training epoch, while

the right ones are for a zoom in the range of $200 < t < 1000$. The red (blue) solid and dashed curves illustrate the values of the cost function for the training (test) sample, before and after averaging, respectively

$f_{stop} > 0$ for more than one hundred iterations, for this example we consider the solution obtained in the epoch 382 as the valid one and we use this solution in the further analysis.

4.5 Feasibility test

In order to check our procedure a feasibility test was performed. In this test the extraction of CFF parameterizations is done on a sample of pseudo-data generated in the

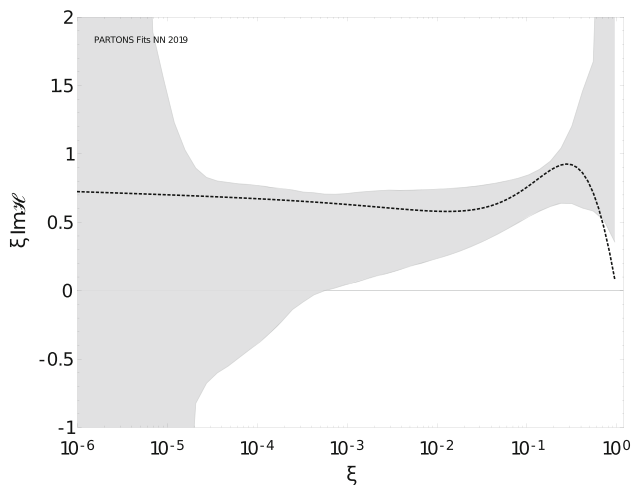


Fig. 10 Example outcome of the feasibility test. The imaginary part of the CFF \mathcal{H} evaluated from GK GPD model [27–29] (dotted line) and the corresponding parameterization of that quantity extracted from the pseudo-data generated with this model (gray band). The plot is shown for $Q^2 = 2 \text{ GeV}^2$ and $t = -0.3 \text{ GeV}^2$

leading-order formalism with the GK GPD model [27–29]. The generation of pseudo-data is straightforward: for each experimental data point, a single point of pseudo-data is generated. This single point of pseudo-data is generated with the same type (differential cross section, or beam spin asymmetry, or beam charge asymmetry, *etc.*), and at the same kinematics as the original point. The obtained observable value is then smeared according to the uncertainties of the original point. As a result of this procedure, pseudo-data are faithfully generated, corresponding to the experimental data; however “input” CFFs are known for this sample. These pseudo-data are used in the same procedure of CFF extraction as the genuine experimental data.

The outcome of this feasibility test is in general positive. The value of the χ^2 function per single data point is 1.13, or 1.06 for data other than HERA and COMPASS points (see Sect. 6 for an explanation of this behavior). The agreement between the CFF parameterizations obtained from the pseudo-data and the GK GPD model used to generate those data is demonstrated with Fig. 10, where the imaginary part of CFF \mathcal{H} and the model are compared as a function of ξ for example kinematics of $t = -0.3 \text{ GeV}^2$ and $Q^2 = 2 \text{ GeV}^2$. The curve representing the model stays within the uncertainty band of the extracted parameterization. The only exception is the region of $\xi \approx 1$, which is not covered by pseudo-data and where the model rapidly goes to zero. However, in this region the difference between the model and the central value of the extracted parameterization is still reasonable (e.g. 1.23σ for $\xi = 0.95$).

5 Experimental data

Table 2 summarizes DVCS data used in this analysis. For the explanation of symbols used to distinguish between observable types see Ref. [14]. Only proton data are used. Available neutron data are sparse ones, but *in principle* they may be used to attempt a flavor separation. This is however beyond the scope of the presented analysis.

As in our previous analysis [14], recent Hall A data [34,35] for cross sections measured with unpolarized beam and target, $d^4\sigma_{UU}^-$, are not used in the final extraction of CFF information. Again, only the corresponding differences of cross sections probing longitudinally polarized beam, $\Delta d^4\sigma_{LU}^-$, are used. We will elaborate on the inclusion of Hall A data in Sect. 6.

The difference with respect to our previous analysis [14] in terms of used data comes from the inclusion of low- x_{Bj} sets. For COMPASS, instead of a single point for the measurement of the t -slope b , four points for the cross section measurement, $d^3\sigma_{UU}^\pm$, are used. The measurement of both slope and cross sections is reported in Ref. [36]. The cross sections provide richer information than the slope itself. In addition, the slope measurement relies on the assumption of the t -dependence being exponential, which can be avoided analyzing the cross section points. However, the COMPASS cross sections are measured in broad kinematic bins and because of that they cannot be compared to models evaluated at the average kinematics of events corresponding to those bins. That is why in this analysis, the COMPASS points are compared to respective integrals of cross sections. Those multidimensional integrals are time consuming and significantly extend the computing time of our fits.

HERA data for DVCS [37–39] are included in this analysis. As we will demonstrate in Sect. 6, those data provide important constraints on CFF parameterizations in the low- x_{Bj} region. However, the sparsity of data covering this region makes our extraction of CFFs difficult, as will be explained in Sect. 6.

We apply two kinematic cuts on experimental data:¹

$$Q^2 > 1.5 \text{ GeV}^2, \quad (22)$$

$$-t/Q^2 < 0.2. \quad (23)$$

The purpose of those cuts is the restriction of the phase-space covered by experimental data to the deeply virtual region, where one can rely on the factorization between GPDs and the hard scattering kernel. *In principle*, those cuts can be avoided in the extractions of amplitudes. However, we keep them here to allow an interpretation of the extracted CFFs in terms of GPDs. In addition, the cuts are applied to keep

¹ There is a typo in Eq. (86) of our last publication [14]. Instead of $-t/Q^2 < 0.25$ there should be $-t/Q^2 < 0.2$, as in Eq. (23) here.

Table 2 DVCS data used in this analysis

No.	Collab.	Year	Ref.	Observable	Kinematic dependence	No. of points used / all
1	HERMES	2001	[40]	A_{LU}^+	ϕ	10/10
2		2006	[41]	$A_C^{\cos i\phi}$	$i = 1$ t	4/4
3		2008	[42]	$A_C^{\cos i\phi}$ $A_{UT,DVCS}^{\sin(\phi-\phi_S)\cos i\phi}$ $A_{UT,I}^{\sin(\phi-\phi_S)\cos i\phi}$ $A_{UT,I}^{\cos(\phi-\phi_S)\sin i\phi}$	$i = 0, 1$ $i = 0$ $i = 0, 1$ $i = 1$	18/24
4		2009	[43]	$A_{LU,I}^{\sin i\phi}$ $A_{LU,DVCS}^{\sin i\phi}$ $A_C^{\cos i\phi}$	$i = 1, 2$ $i = 1$ $i = 0, 1, 2, 3$	x_{Bj} 35/42
5		2010	[44]	$A_{UL}^{+,\sin i\phi}$ $A_{LL}^{+,\cos i\phi}$	$i = 1, 2, 3$ $i = 0, 1, 2$	x_{Bj} 18/24
6		2011	[45]	$A_{LT,DVCS}^{\cos(\phi-\phi_S)\cos i\phi}$ $A_{LT,DVCS}^{\sin(\phi-\phi_S)\sin i\phi}$ $A_{LT,I}^{\cos(\phi-\phi_S)\cos i\phi}$ $A_{LT,I}^{\sin(\phi-\phi_S)\sin i\phi}$	$i = 0, 1$ $i = 1$ $i = 0, 1, 2$ $i = 1, 2$	x_{Bj} 24/32
7		2012	[46]	$A_{LU,I}^{\sin i\phi}$ $A_{LU,DVCS}^{\sin i\phi}$ $A_C^{\cos i\phi}$	$i = 1, 2$ $i = 1$ $i = 0, 1, 2, 3$	x_{Bj} 35/42
8	CLAS	2001	[47]	$A_{LU}^{-,\sin i\phi}$	$i = 1, 2$ –	0/2
9		2006	[48]	$A_{UL}^{-,\sin i\phi}$	$i = 1, 2$ –	2/2
10		2008	[49]	A_{LU}^-	ϕ	283/737
11		2009	[50]	A_{LU}^-	ϕ	22/33
12		2015	[51]	$A_{LU}^-, A_{UL}^-, A_{LL}^-$	ϕ	311/497
13		2015	[52]	$d^4\sigma_{UU}^-$	ϕ	1333/1933
14	Hall A	2015	[34]	$\Delta d^4\sigma_{LU}^-$	ϕ	228/228
15		2017	[35]	$\Delta d^4\sigma_{LU}^-$	ϕ	276/358
16	COMPASS	2018	[36]	$d^3\sigma_{UU}^\pm$	t	2/4
17	ZEUS	2009	[37]	$d^3\sigma_{UU}^+$	t	4/4
18	H1	2005	[38]	$d^3\sigma_{UU}^+$	t	7/8
19		2009	[39]	$d^3\sigma_{UU}^\pm$	t	12/12
SUM:						2624/3996

a correspondence with our previous analysis [14], allowing a straightforward comparison. Let us note at this point, that the neural network approach developed in this work provides an easy way for the addition of CFFs identified with higher-twist contributions. With those contributions included one could relax the cuts (22) and (23), allowing more data to be included.

In total, in this analysis 2624 points are used out of 3996 available in all considered data sets. The coverage of phase-space by those data in the sections of (x_{Bj}, Q^2) and $(x_{Bj}, -t/Q^2)$ is shown in Fig. 11. With those plots one can easily conclude about the actual coverage by various exper-

iments, but also identify parts of phase-space not probed at all by existing data.

6 Results

6.1 Performance

The value of the χ^2 function obtained after the training of our neural network system with 2624 experimental points is 2243.5 for the central replica. It gives the average deviation of that system’s answer from experimental data $2243.5/2624 \approx$

Fig. 11 Coverage of the (x_{Bj}, Q^2) (left) and $(x_{\text{Bj}}, -t/Q^2)$ (right) phase-spaces by the experimental data listed in Table 2. The data come from the Hall A (∇, \triangledown), CLAS ($\blacktriangle, \triangle$), HERMES (\bullet, \circ), COMPASS (\blacksquare, \square) and HERA H1 and ZEUS ($\blacklozenge, \blacklozenge$) experiments. The gray bands (open markers) indicate phase-space areas (experimental points) being excluded from this analysis due to the cuts introduced in Eqs. (22) and (23)

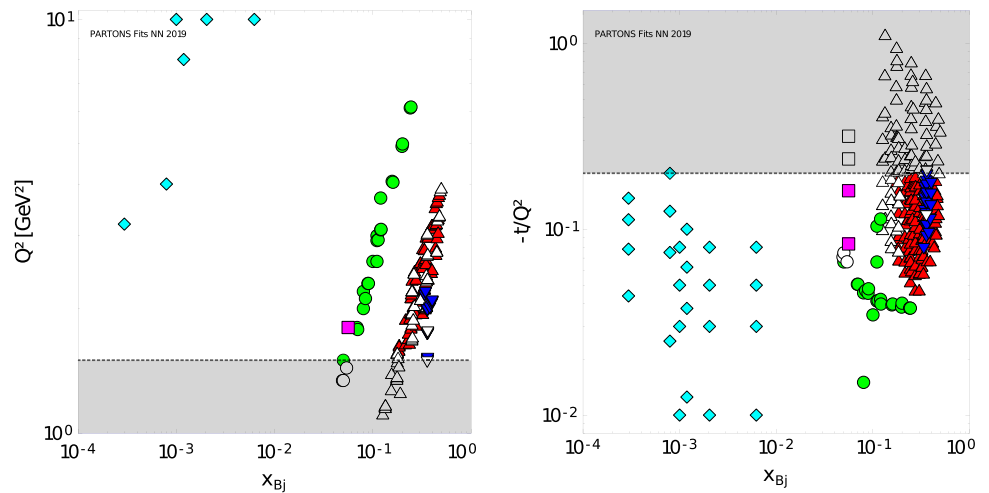


Table 3 Values of the χ^2 function per data set. For a given data set, cf. Table 2, given are: χ^2 value, the number of experimental points n , and the ratio between those two numbers

No.	Collab.	Year	Ref.	χ^2	n	χ^2/n
1	HERMES	2001	[40]	10.7	10	1.07
2		2006	[41]	5.5	4	1.38
3		2008	[42]	18.5	18	1.03
4		2009	[43]	34.7	35	0.99
5		2010	[44]	40.7	18	2.26
6		2011	[45]	16.7	24	0.70
7		2012	[46]	22.4	35	0.64
8	CLAS	2001	[47]	–	0	–
9		2006	[48]	1.0	2	0.52
10		2008	[49]	376.4	283	1.33
11		2009	[50]	28.3	22	1.29
12		2015	[51]	306.6	311	0.99
13		2015	[52]	884.7	1333	0.66
14	Hall A	2015	[34]	231.8	228	1.02
15		2017	[35]	211.4	276	0.77
16	COMPASS	2018	[36]	3.0	2	1.50
17	ZEUS	2009	[37]	5.49	4	1.38
18	H1	2005	[38]	22.2	7	3.17
19		2009	[39]	23.4	12	1.95

0.85 per single data point. This value per experimental data set is given in Table 3. The number of free parameters, that is the number of weights and biases in all networks, is 248. There is not much sense of studying the goodness of fit with this information taken into account, as we know *a priori* that the size of a single network may be too large, and that is why we are using a regularization. In other words, we are able to increase the number of neurons in our network, and therefore the number of weight and biases, keeping at the same time the same precision of data description.

Figures 12, 13, 14 and 15 provide a straightforward comparison between our fit and the selected data sets. As indicated by the χ^2 values summarized in Table 3, we are able to describe the data well within a single phenomenological framework based on the neural network approach. This includes data ranging from HERA to JLab kinematics. Beyond the results of this analysis, predictions coming from the GK [27–29] and VGG [53–56] GPD models are also shown. Those two models originate from the exploratory phase of GPD studies and are able to describe only a general behavior followed by experimental data. This confirms the need for new GPD models constrained in global analyses, preferably multi-channel ones.

As has been already mentioned in Sect. 5, Hall A cross sections are not included in the nominal data set that is used in this analysis. We point out, that in our last analysis [14] we were not able to describe those cross sections with the proposed Ansatz and those data were excluded there as well. The question is: can those data be described in this analysis, where a flexible CFF parameterization based on the neural network approach is used? The answer consists of two parts: (i) the parameterizations of CFFs obtained in this analysis from the nominal data set do not have the predictive power to describe Hall A cross sections. The value of the χ^2 function for those cross sections is 5916.6, which for 594 points gives the reduced value of 9.96. The bad description is also seen by eye in Fig. 13. (ii) the inclusion of Hall A cross sections in the extraction of CFFs significantly improves the χ^2 evaluated for this subset, from aforementioned 9.96 to 1.12 per single data point. However, such an inclusion makes the χ^2 for the CLAS cross sections worse, from 0.66 reported in Table 3 to 0.83 per single data point. Taking both observations into account we found the situation unclear at this moment. Hall A cross sections will be investigated further in a future analysis with higher-twist contributions taken into account.

The inclusion of low- x_{Bj} experimental data has significantly extended the coverage of phase-space. However, the

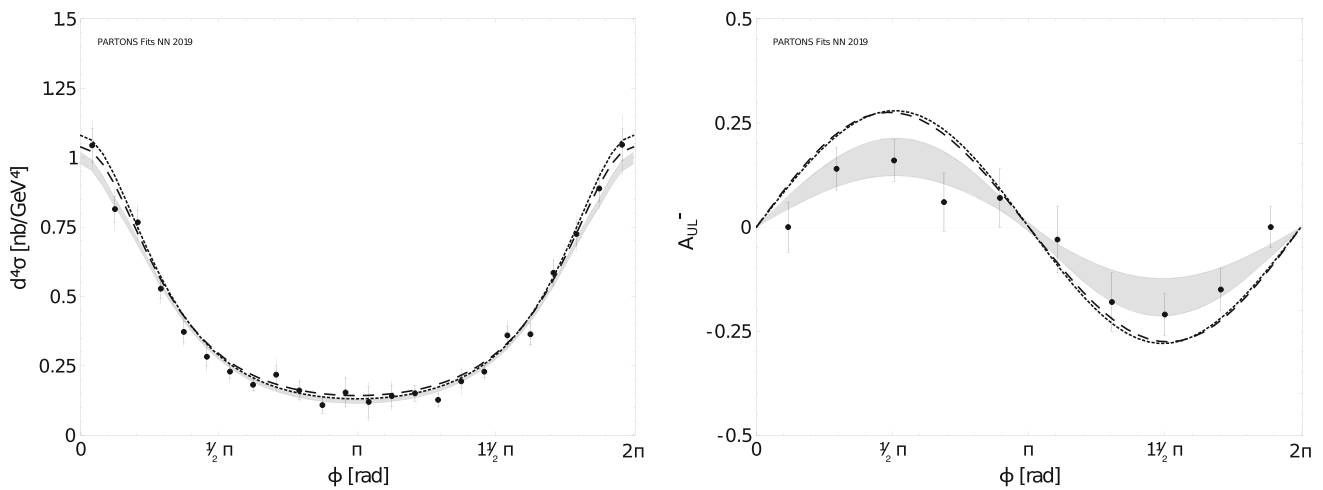


Fig. 12 Comparison between the results of this analysis, the selected GPD models and experimental data published by CLAS in Refs. [51, 52] for $d^4\sigma_{UU}^-$ at $x_{Bj} = 0.244$, $t = -0.15 \text{ GeV}^2$ and $Q^2 = 1.79 \text{ GeV}^2$ (left) and for A_{UL}^- at $x_{Bj} = 0.2569$, $t = -0.23 \text{ GeV}^2$, $Q^2 = 2.019 \text{ GeV}^2$ (right). The gray bands correspond to the results of this analysis with

68% confidence level for the uncertainties coming from DVCS data, respectively. The dotted curve is for the GK GPD model [27–29], while the dashed one is for VGG [53–56]. The curves are evaluated at the kinematics of experimental data

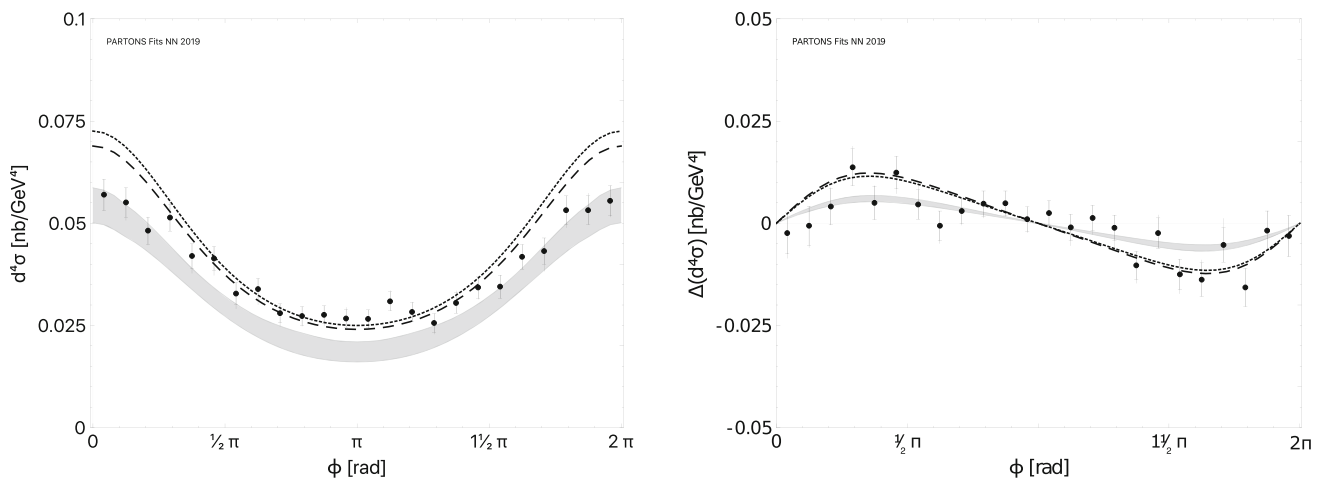


Fig. 13 Comparison between the results of this analysis, the selected GPD models and experimental data published by Hall A in Ref. [34] for $d^4\sigma_{UU}^-$ (left) and $\Delta d^4\sigma_{LU}$ (right) at $x_{Bj} = 0.392$, $t = -0.233 \text{ GeV}^2$ and $Q^2 = 2.054 \text{ GeV}^2$. For further description see the caption of Fig. 12

sparsity of those data creates problems in the supervised training that we utilize in this analysis, see Sect. 3.3. Namely, the random division of the available experimental data into training and test subsets causes an insufficient coverage of some parts of the phase-space by either the training or the test points. This may lead to over-fitting, if a given part of the phase-space is only covered by training points, or under-fitting, if kinematics is only covered by test points. The effect increases the spread of replicas and because of that gives an additional contribution to the estimated uncertainties. The goodness of fit for those data is also worse as shown in Table 3. Foreseen data coming from electron-ion facilities will improve this situation.

6.2 Compton form factors

The parameterizations of CFFs are shown in Figs. 16, 17, 18 and 19 for example kinematics of $t = -0.3 \text{ GeV}^2$ and $Q^2 = 2 \text{ GeV}^2$ as a function of ξ . As expected the data provide the best constraints on $\text{Im}\mathcal{H}$, and some on $\text{Re}\mathcal{H}$, $\text{Im}\tilde{\mathcal{H}}$ and $\text{Re}\tilde{\mathcal{E}}$. Other CFFs are poorly constrained by the available data, in particular \mathcal{E} related to GPD E , being of a great importance for the study of parton densities in a transversely polarized proton and the determination of the orbital angular momentum through Ji’s sum rule.

The inclusion of HERA and COMPASS data in a global extraction of CFFs is not trivial, but those data provide impor-

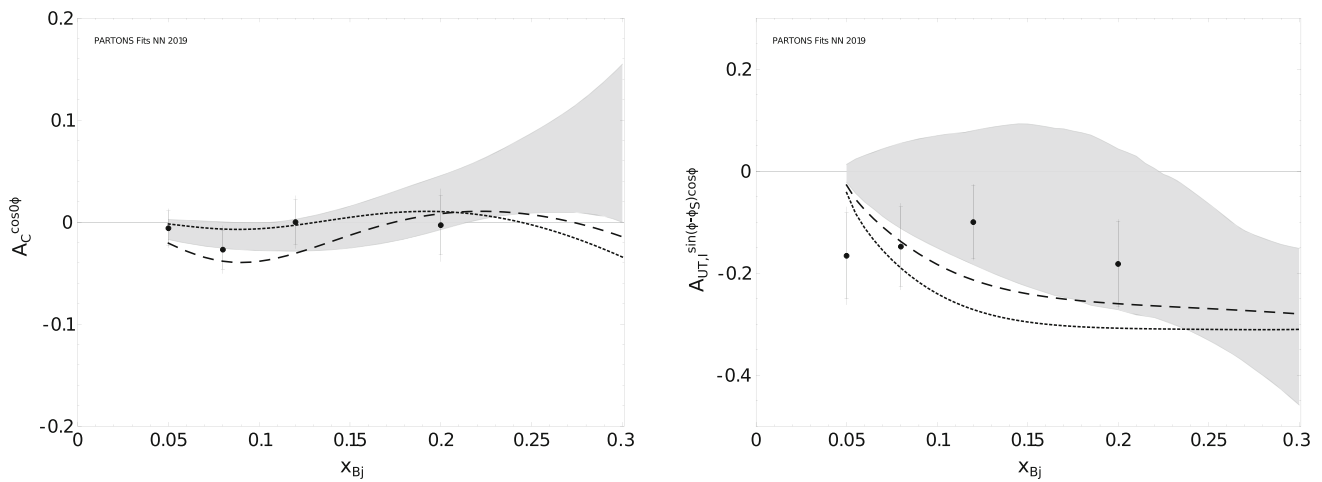


Fig. 14 Comparison between the results of this analysis, the selected GPD models and the experimental data published by HERMES in Ref. [42] for $A_C^{\cos 0\phi}$ (left) and $A_{UT,1}^{\sin(\phi-\phi_S)\cos\phi}$ (right) at $t = -0.12 \text{ GeV}^2$ and $Q^2 = 2.5 \text{ GeV}^2$. For further description see the caption of Fig. 12

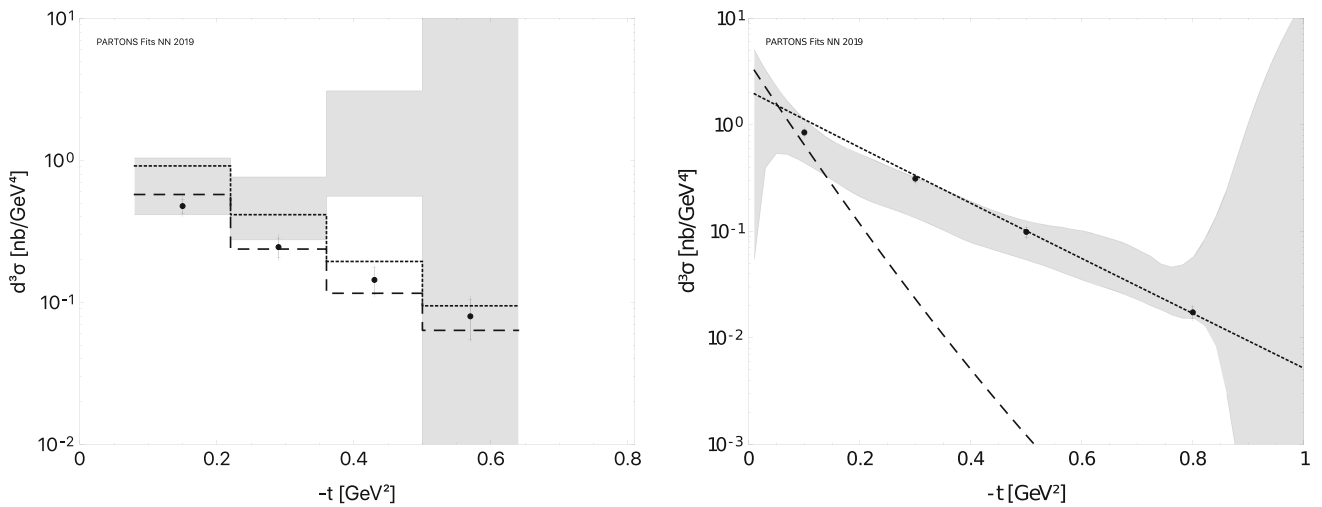


Fig. 15 Comparison between the results of this analysis, the selected GPD models and the experimental data published by COMPASS in Ref. [36] for $d^3\sigma_{UU^\pm}$ (left) and by H1 in Ref. [39] for $d^3\sigma_{UU^+}$ at $x_{Bj} = 0.002$ and $Q^2 = 10 \text{ GeV}^2$ (right). In the left plot, the experimental data, the

results of this analysis and the GPD model evaluations are 3D integrals in four bins of the (ν, Q^2, t) phase-space, where ν is the virtual-photon energy. The ranges of those bins are specified in Ref. [36]. For further description see the caption of Fig. 12

tant constraints in the low and intermediate range of x_{Bj} . We demonstrate it with the example of Fig. 20, where the imaginary part of the CFF \mathcal{H} is shown without the inclusion of low- x_{Bj} data. We point out that the gap between the collider and fixed target experiments seen in the coverage of the (x_{Bj}, Q^2) phase-space in Fig. 11 is expected to be filled by future experiments at expected electron-ion collider facilities. The precision of foreseen data should allow for a precise phenomenology in that domain.

The Q^2 evolution of the CFF \mathcal{H} for $\xi = 0.002$ and $\xi = 0.2$ is shown in Fig. 21. One can note a rather mild Q^2 -dependence followed by the extracted CFF parameterizations, consistent with the expected logarithmic behavior.

6.3 Nucleon tomography

In Fig. 22 we show our results for the slope b of the DVCS cross section described by a single exponential function, $d^3\sigma_{UU} \propto \exp(bt)$. This slope is evaluated as indicated in our previous analysis [14]. It can be converted into a transverse extension of partons under assumptions that are also specified in Ref. [14]. The uncertainties on the extracted values of b are larger than expected, mainly because of the aforementioned problems with the supervised training caused by the sparsity of the low- x_{Bj} data. In addition, without an explicit assumption about the exponential t -behavior of $\text{Im}\mathcal{H}$, the estimation of b from pliant replicas gets additional uncertainties.

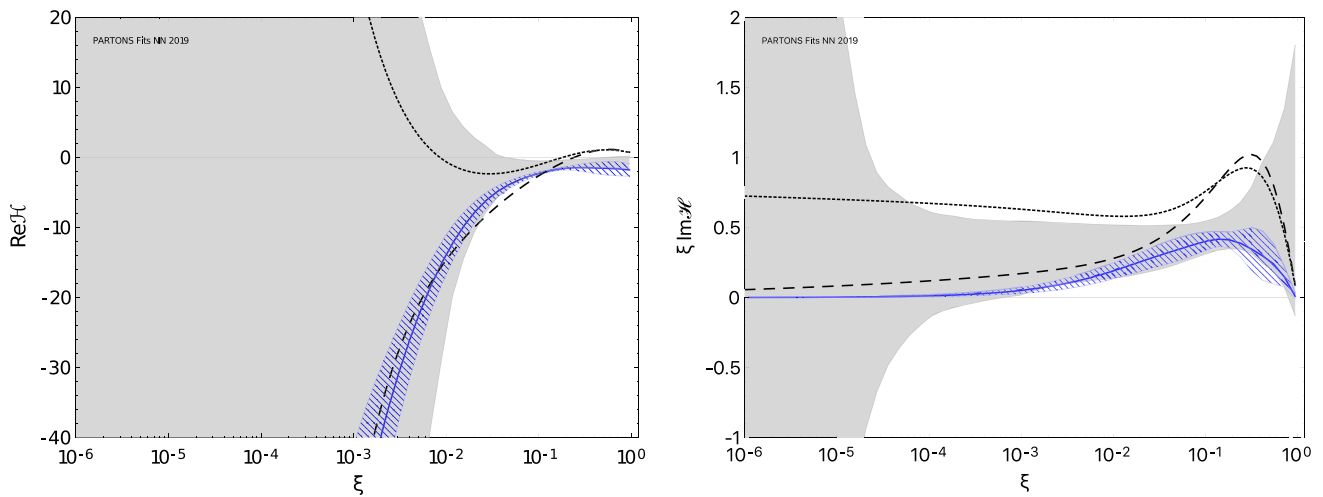


Fig. 16 Real (left) and imaginary (right) parts of the CFF \mathcal{H} as a function of ξ for $t = -0.3 \text{ GeV}^2$ and $Q^2 = 2 \text{ GeV}^2$. The blue solid line surrounded by the blue hatched band denotes the result of our previous analysis [14]. The depicted uncertainty accounts for uncertainties

estimated in that analysis for experimental data, unpolarized, polarized PDFs and elastic form factors. For further description see the caption of Fig. 12

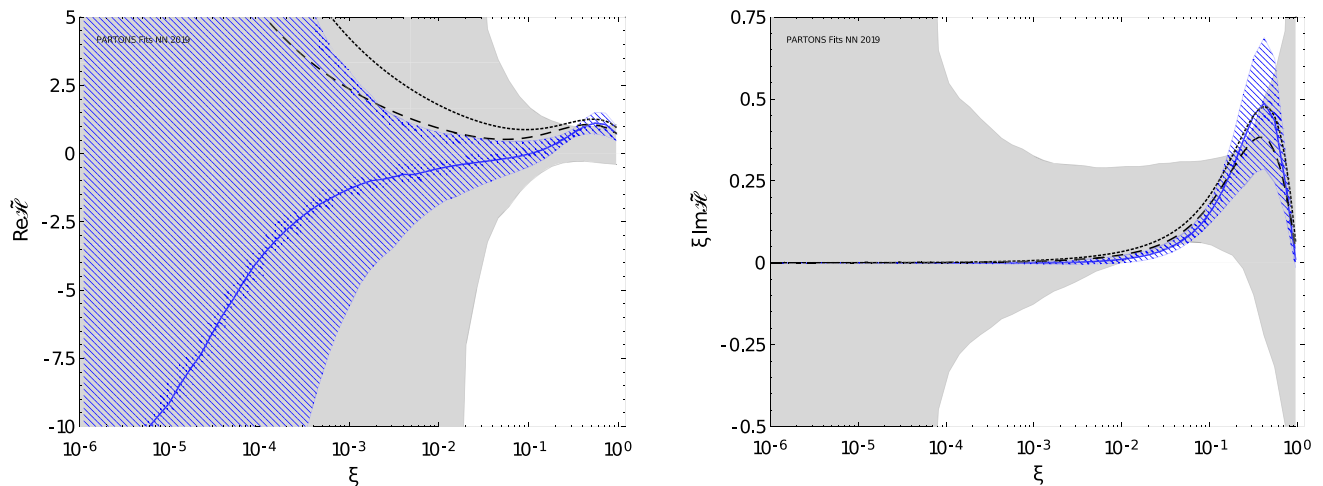


Fig. 17 Real (left) and imaginary (right) parts of the CFF $\tilde{\mathcal{H}}$ as a function of ξ for $t = -0.3 \text{ GeV}^2$ and $Q^2 = 2 \text{ GeV}^2$. For further description see the caption of Fig. 16

The extraction of tomography information from CFFs is also possible in the high- x_{Bj} range, see for instance Refs. [32,33]. However, this requires a “de-skewing” of $\text{Im}\mathcal{H}$, *i.e.* one needs to evaluate the GPD $H^q(x, 0, t)$ from:

$$\text{Im}\mathcal{H}(\xi, t) \stackrel{\text{LO}}{=} \pi \sum_q e_q^2 H^{q(+)}(x = \xi, \xi, t), \tag{24}$$

where the sum runs over quark flavors q , e_q is the electric charge of a specific quark flavor in units of the positron charge e and where $H^{q(+)}(x, \xi, t) = H^q(x, \xi, t) - H^q(-x, \xi, t)$. Because of the model-dependency of this procedure we refrain from doing it here. We point out that a straightforward, however still model-dependent, access to the nucleon

tomography is a feature of the CFF Ansätze proposed in our previous analysis [14].

6.4 Subtraction constant

The subtraction constant C_H is evaluated with our results on the CFF \mathcal{H} and the dispersion relation introduced in Sect. 2. More precisely, the imaginary part of the CFF \mathcal{H} is integrated according to Eq. (5) in the range of $\epsilon < \xi' < 1$ and then subtracted from the corresponding real part. By studying C_H as a function of ϵ the latter has been chosen to be 10^{-6} , which introduces a little bias on C_H comparing to $\epsilon = 0$. This bias is estimated to be smaller than 1%.

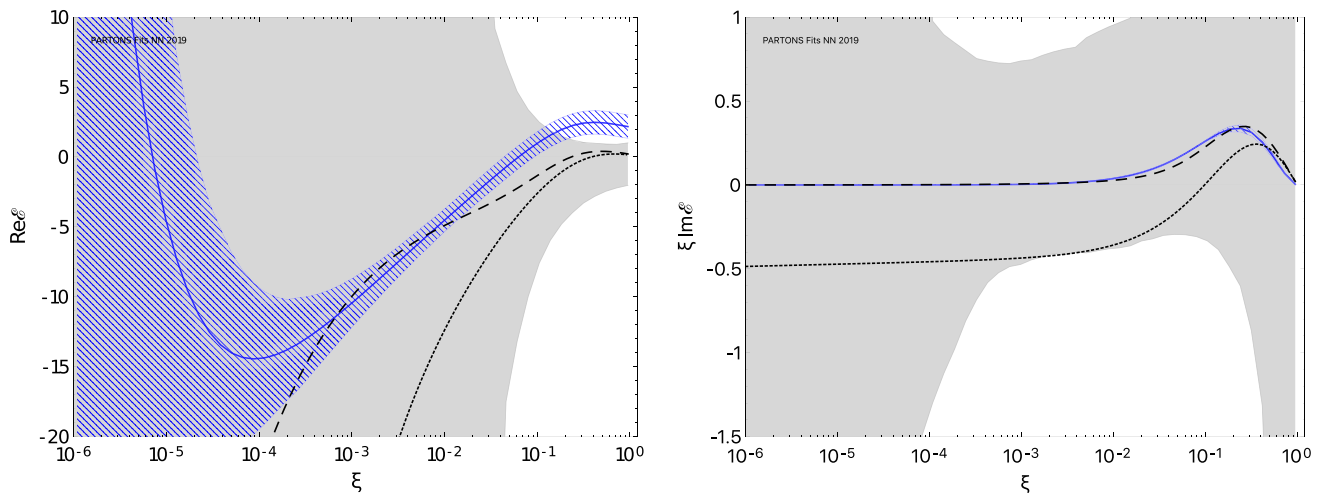


Fig. 18 Real (left) and imaginary (right) parts of the CFF \mathcal{E} as a function of ξ for $t = -0.3 \text{ GeV}^2$ and $Q^2 = 2 \text{ GeV}^2$. For further description see the caption of Fig. 16

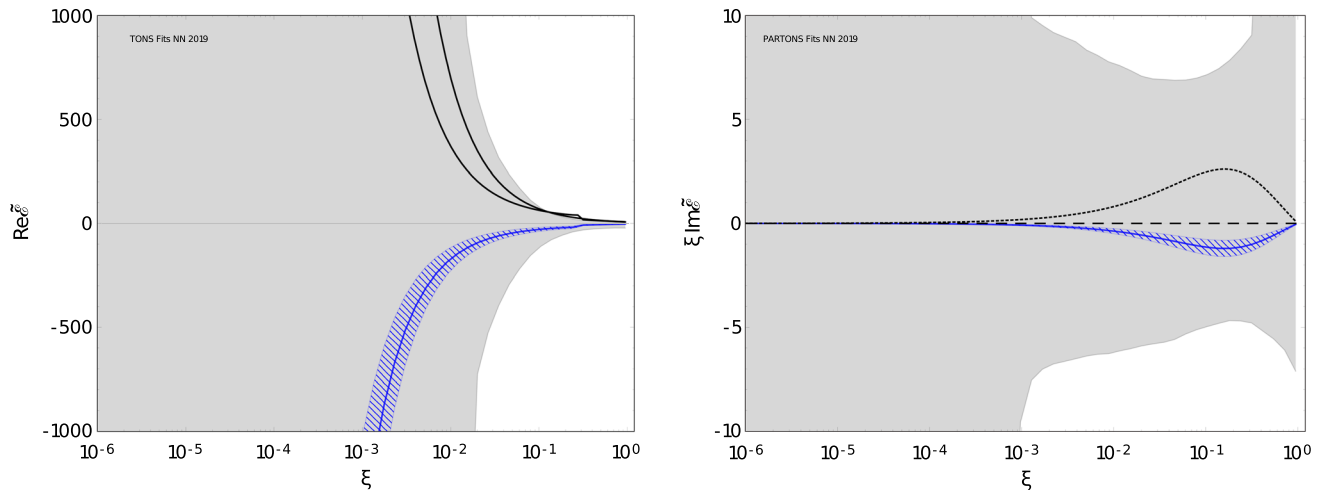


Fig. 19 Real (left) and imaginary (right) parts of the CFF $\tilde{\mathcal{E}}$ as a function of ξ for $t = -0.3 \text{ GeV}^2$ and $Q^2 = 2 \text{ GeV}^2$. For further description see the caption of Fig. 16

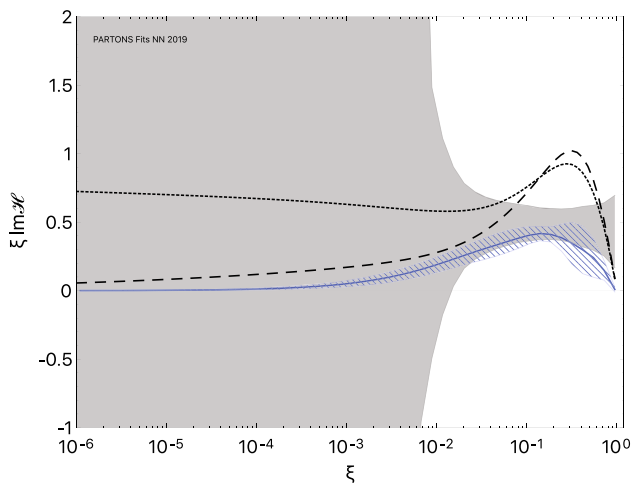


Fig. 20 The same as Fig. 16, but without the HERA and COMPASS data taken into account in the extraction of CFFs

The subtraction constant for a given Q^2 and t should be the same independently on the value of ξ used in its extraction. This is demonstrated in Fig. 23 (top), where C_H is shown as a function of ξ for example kinematics of $t = -0.3 \text{ GeV}^2$ and $Q^2 = 2 \text{ GeV}^2$. We consider this test as a proof of consistency between the parameterizations of the real and imaginary parts of the CFF \mathcal{H} .

The subtraction constant as a function of t and Q^2 is shown in Fig. 23 (bottom) for $\xi = 0.2$. As expected, the uncertainties are large in domains sparsely covered by data, in particular for $t \rightarrow 0$ and $Q^2 \rightarrow \infty$, which are important for a direct interpretation of C_H in terms of the energy-momentum tensor. However, the uncertainties obtained in domains covered by data are encouraging, so our result coming from a model-independent extraction can provide important constraints on models of C_H . Our findings are consistent with the recent observation of Ref. [16].

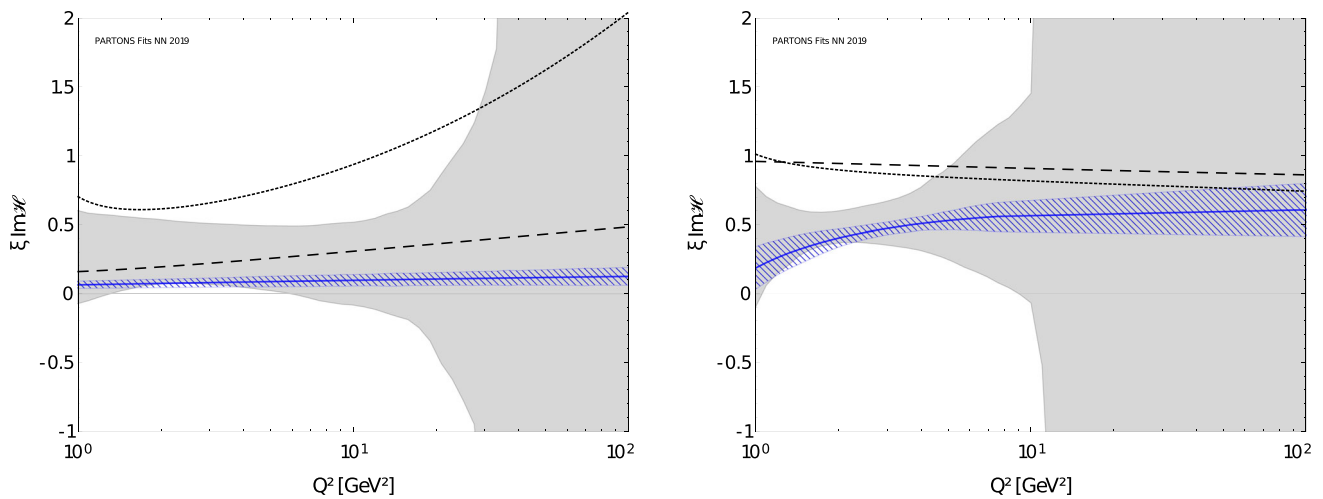


Fig. 21 Imaginary part of the CFF \mathcal{H} as a function of Q^2 for $t = -0.3 \text{ GeV}^2$ and $\xi = 0.002$ (left) and $\xi = 0.2$ (right). For further description see the caption of Fig. 16

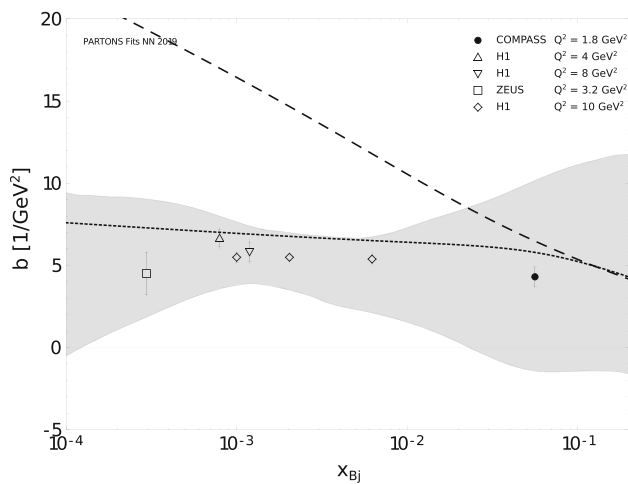


Fig. 22 Comparison between the results of this analysis, the selected GPD models and the experimental data by the COMPASS [36], ZEUS [37] and H1 [38,39] Collaborations for the slope b at $Q^2 = 10 \text{ GeV}^2$. Note that the shown data differ by Q^2 values indicated in the legend. For further description see the caption of Fig. 12

7 Summary

In this paper we report the extraction of CFF parameterizations from proton DVCS data. The extracted quantities are the most basic observables as one can unambiguously access by exploring the DVCS process. We analyze the data in the region in which the interpretation in the language of GPDs is applicable. In this analysis, a dispersion relation is used to access the DVCS subtraction constant, which is related to the mechanical forces acting on partons in the nucleon. Also, the nucleon tomography that describes a spatial distribution of partons inside the nucleon is discussed.

The extraction of CFFs is done with the help of the artificial neural networks technique, allowing for an essential reduction of model dependency. The presented analysis includes such elements as the training of neural networks with the genetic algorithm, the careful regularization to avoid over-fitting and the propagation of experimental uncertainties with the replica method. The work is done within the PARTONS framework [9].

The results of this analysis include in particular unbiased CFF parameterizations extracted in the three-dimensional phase-space of (x_{Bj}, t, Q^2) , with a reliable estimation of uncertainties. In addition, a direct extraction of the subtraction constant from the experimental data is presented.

The analysis is complementary to our previous one [14], where CFF parameterizations were constructed with the basic GPD properties acting as Ansatz building blocks. Although a physically motivated Ansatz gives more insight into GPD physics, its inherent model-dependency introduces an extra theoretical uncertainty, which usually can be only roughly estimated. The situation is opposite for the analysis presented in this paper – the extraction of CFFs is unbiased, but the link with GPD physics is not as straightforward as in the case of previous analysis. Therefore, both analyses, which are performed on a similar data set, provide a complete picture of the DVCS process.

This work provides a benchmark for a powerful tool to be used in future GPD analyses. This tool allows in particular for a nearly model-independent estimation of the impact of future experiments, in particular those to be performed in the foreseen electron-ion facilities. CFF parameterizations that are presented in this paper can be used for a fast generation of DVCS cross sections, which may be useful e.g. for Monte Carlo generation. Studies of Timelike Compton Scattering

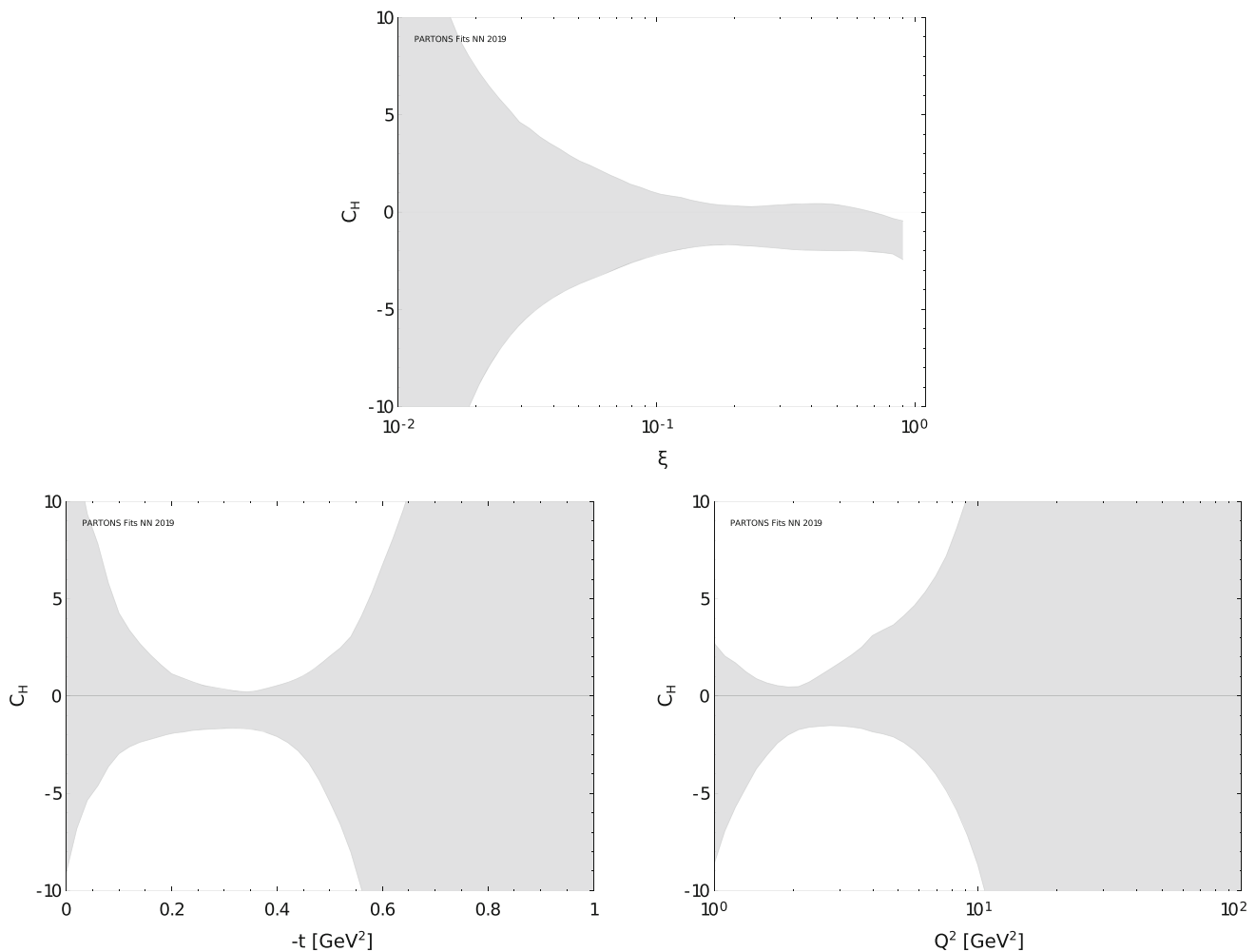


Fig. 23 Subtraction constant evaluated from the CFF \mathcal{H} as a function of ξ for $t = -0.3 \text{ GeV}^2$ and $Q^2 = 2 \text{ GeV}^2$ (top), as a function of $-t$ for $\xi = 0.2$ and $Q^2 = 2 \text{ GeV}^2$ (bottom left) and as a function of Q^2 for $\xi = 0.2$ and $t = -0.3 \text{ GeV}^2$ (bottom right)

and higher twist contributions are also possible within the proposed framework.

Acknowledgements The authors would like to thank C. Weiss and K. Kumerički for fruitful discussions. This work was supported by the Grant No. 2017/26/M/ST2/01074 of the National Science Centre, Poland and was done in part under CRADA Agreement No. 2017S002 between Jefferson Lab, USA and National Centre for Nuclear Research, Poland. The project is co-financed by the Polish National Agency for Academic Exchange and by the COPIN-IN2P3 Agreement. The computing resources of Świerk Computing Centre, Poland are greatly acknowledged.

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: The obtained parameterisations of CFFs will be publicly available in the next release of the PARTONS framework.]

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative

Commons license, and indicate if changes were made. Funded by SCOAP³.

References

1. M. Diehl, Eur. Phys. J. A **52**(6), 149 (2016). <https://doi.org/10.1140/epja/i2016-16149-3>
2. A. Bacchetta, Eur. Phys. J. A **52**(6), 163 (2016). <https://doi.org/10.1140/epja/i2016-16163-5>
3. X.D. Ji, Phys. Rev. Lett. **78**, 610 (1997). <https://doi.org/10.1103/PhysRevLett.78.610>
4. X.D. Ji, Phys. Rev. D **55**, 7114 (1997). <https://doi.org/10.1103/PhysRevD.55.7114>
5. M.V. Polyakov, A.G. Shuvaev. On 'dual' parametrizations of generalized parton distributions (2002). [arXiv:hep-ph/0207153](https://arxiv.org/abs/hep-ph/0207153)
6. M.V. Polyakov, Phys. Lett. B **555**, 57 (2003). [https://doi.org/10.1016/S0370-2693\(03\)00036-4](https://doi.org/10.1016/S0370-2693(03)00036-4)
7. M.V. Polyakov, P. Schweitzer, Int. J. Mod. Phys. A **33**(26), 1830025 (2018). <https://doi.org/10.1142/S0217751X18300259>
8. C. Lorcé, H. Moutarde, A.P. Trawiński, Eur. Phys. J. C **79**(1), 89 (2019). <https://doi.org/10.1140/epjc/s10052-019-6572-3>

9. B. Berthou et al., Eur. Phys. J. C **78**(6), 478 (2018). <https://doi.org/10.1140/epjc/s10052-018-5948-0>
10. K. Kumericki, S. Liuti, H. Moutarde, Eur. Phys. J. A **52**(6), 157 (2016). <https://doi.org/10.1140/epja/i2016-16157-3>
11. N. d'Hose, S. Niccolai, A. Rostomyan, Eur. Phys. J. A **52**(6), 151 (2016). <https://doi.org/10.1140/epja/i2016-16151-9>
12. M.H. Hassoun, *Fundamentals of Artificial Neural Networks* (MIT Press, Cambridge, 1995)
13. K. Kumericki, D. Mueller, A. Schafer, JHEP **07**, 073 (2011). [https://doi.org/10.1007/JHEP07\(2011\)073](https://doi.org/10.1007/JHEP07(2011)073)
14. H. Moutarde, P. Sznajder, J. Wagner, Eur. Phys. J. C **78**(11), 890 (2018). <https://doi.org/10.1140/epjc/s10052-018-6359-y>
15. V.D. Burkert, L. Elouadrhiri, F.X. Girod, Nature **557**(7705), 396 (2018). <https://doi.org/10.1038/s41586-018-0060-z>
16. K. Kumerički, Nature **570**(7759), E1 (2019). <https://doi.org/10.1038/s41586-019-1211-6>
17. P. Kroll, M. Schurmann, P.A.M. Guichon, Nucl. Phys. A **598**, 435 (1996). [https://doi.org/10.1016/0375-9474\(96\)00002-4](https://doi.org/10.1016/0375-9474(96)00002-4)
18. A.V. Belitsky, D. Müller, Y. Ji, Nucl. Phys. B **878**, 214 (2014). <https://doi.org/10.1016/j.nuclphysb.2013.11.014>
19. M. Diehl, DYu. Ivanov, Eur. Phys. J. C **52**, 919 (2007). <https://doi.org/10.1140/epjc/s10052-007-0401-9>
20. M. Burkardt, Phys. Rev. D **62**, 071503 (2000). <https://doi.org/10.1103/PhysRevD.62.071503>. <https://doi.org/10.1103/PhysRevD.66.119903>. [Erratum: Phys. Rev. D **66**, 119903 (2002)]
21. M. Burkardt, Int. J. Mod. Phys. A **18**, 173 (2003). <https://doi.org/10.1142/S0217751X03012370>
22. M. Burkardt, Phys. Lett. B **595**, 245 (2004). <https://doi.org/10.1016/j.physletb.2004.05.070>
23. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning. Adaptive Computation and Machine Learning Series* (MIT Press, Cambridge, 2016)
24. M. Mitchell, *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, 1998)
25. L. Prechelt, *Early Stopping—But When?* (Springer, Berlin, 2012), pp. 53–67. https://doi.org/10.1007/978-3-642-35289-8_5
26. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, J. Mach. Learn. Res. **15**(1), 1929 (2014)
27. S.V. Goloskokov, P. Kroll, Eur. Phys. J. C **42**, 281 (2005). <https://doi.org/10.1140/epjc/s2005-02298-5>
28. S.V. Goloskokov, P. Kroll, Eur. Phys. J. C **53**, 367 (2008). <https://doi.org/10.1140/epjc/s10052-007-0466-5>
29. S.V. Goloskokov, P. Kroll, Eur. Phys. J. C **65**, 137 (2010). <https://doi.org/10.1140/epjc/s10052-009-1178-9>
30. S. Nissen, *Implementation of a Fast Artificial Neural Network Library (fann)*. Tech. rep., Department of Computer Science University of Copenhagen (DIKU) (2003). <http://fann.sf.net>
31. S.E. Fahlman, *Proceedings of the 1988 Connectionist Models Summer School* (Morgan Kaufmann, San Francisco, CA, 1989), p. 38
32. R. Dupre, M. Guidal, M. Vanderhaeghen, Phys. Rev. D **95**(1), 011501 (2017). <https://doi.org/10.1103/PhysRevD.95.011501>
33. R. Dupré, M. Guidal, S. Niccolai, M. Vanderhaeghen, Eur. Phys. J. A **53**(8), 171 (2017). <https://doi.org/10.1140/epja/i2017-12356-8>
34. M. Defurne et al., Phys. Rev. C **92**(5), 055202 (2015). <https://doi.org/10.1103/PhysRevC.92.055202>
35. M. Defurne et al., Nat. Commun. **8**(1), 1408 (2017). <https://doi.org/10.1038/s41467-017-01819-3>
36. R. Akhunzyanov et al., Phys. Lett. B **793**, 188 (2019). <https://doi.org/10.1016/j.physletb.2019.04.038>
37. S. Chekanov et al., JHEP **05**, 108 (2009). <https://doi.org/10.1088/1126-6708/2009/05/108>
38. A. Aktas et al., Eur. Phys. J. C **44**, 1 (2005). <https://doi.org/10.1140/epjc/s2005-02345-3>
39. F.D. Aaron et al., Phys. Lett. B **681**, 391 (2009). <https://doi.org/10.1016/j.physletb.2009.10.035>
40. A. Airapetian et al., Phys. Rev. Lett. **87**, 182001 (2001). <https://doi.org/10.1103/PhysRevLett.87.182001>
41. A. Airapetian et al., Phys. Rev. D **75**, 011103 (2007). <https://doi.org/10.1103/PhysRevD.75.011103>
42. A. Airapetian et al., JHEP **06**, 066 (2008). <https://doi.org/10.1088/1126-6708/2008/06/066>
43. A. Airapetian et al., JHEP **11**, 083 (2009). <https://doi.org/10.1088/1126-6708/2009/11/083>
44. A. Airapetian et al., JHEP **06**, 019 (2010). [https://doi.org/10.1007/JHEP06\(2010\)019](https://doi.org/10.1007/JHEP06(2010)019)
45. A. Airapetian et al., Phys. Lett. B **704**, 15 (2011). <https://doi.org/10.1016/j.physletb.2011.08.067>
46. A. Airapetian et al., JHEP **07**, 032 (2012). [https://doi.org/10.1007/JHEP07\(2012\)032](https://doi.org/10.1007/JHEP07(2012)032)
47. S. Stepanyan et al., Phys. Rev. Lett. **87**, 182002 (2001). <https://doi.org/10.1103/PhysRevLett.87.182002>
48. S. Chen et al., Phys. Rev. Lett. **97**, 072002 (2006). <https://doi.org/10.1103/PhysRevLett.97.072002>
49. F.X. Girod et al., Phys. Rev. Lett. **100**, 162002 (2008). <https://doi.org/10.1103/PhysRevLett.100.162002>
50. G. Gavalian et al., Phys. Rev. C **80**, 035206 (2009). <https://doi.org/10.1103/PhysRevC.80.035206>
51. S. Pisano et al., Phys. Rev. D **91**(5), 052014 (2015). <https://doi.org/10.1103/PhysRevD.91.052014>
52. H.S. Jo et al., Phys. Rev. Lett. **115**(21), 212003 (2015). <https://doi.org/10.1103/PhysRevLett.115.212003>
53. M. Vanderhaeghen, P.A.M. Guichon, M. Guidal, Phys. Rev. Lett. **80**, 5064 (1998). <https://doi.org/10.1103/PhysRevLett.80.5064>
54. M. Vanderhaeghen, P.A.M. Guichon, M. Guidal, Phys. Rev. D **60**, 094017 (1999). <https://doi.org/10.1103/PhysRevD.60.094017>
55. K. Goeke, M.V. Polyakov, M. Vanderhaeghen, Prog. Part. Nucl. Phys. **47**, 401 (2001). [https://doi.org/10.1016/S0146-6410\(01\)00158-2](https://doi.org/10.1016/S0146-6410(01)00158-2)
56. M. Guidal, M.V. Polyakov, A.V. Radyushkin, M. Vanderhaeghen, Phys. Rev. D **72**, 054013 (2005). <https://doi.org/10.1103/PhysRevD.72.054013>